

**UNIVERSIDADE FEDERAL DE SANTA MARIA  
CENTRO DE TECNOLOGIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA**

**João Otávio Cadó de Matos**

**OTIMIZAÇÃO DO MÉTODO DE LOCALIZAÇÃO DE MONTE CARLO  
PARA REDES DE SENSORES SEM FIO MÓVEIS**

Santa Maria, RS  
2016



**João Otávio Cadó de Matos**

**OTIMIZAÇÃO DO MÉTODO DE LOCALIZAÇÃO DE MONTE CARLO PARA  
REDES DE SENSORES SEM FIO MÓVEIS**

Dissertação apresentada ao Curso de Mestrado do Programa de Pós-Graduação em Informática (PPGI), área de concentração em Computação, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Mestre em Informática**.

Orientador: Prof<sup>o</sup>. Dr. João Baptista dos Santos Martins

Santa Maria, RS  
2016

Ficha catalográfica elaborada através do Programa de Geração Automática da Biblioteca Central da UFSM, com os dados fornecidos pelo(a) autor(a).

Matos, João Otávio Cadó de  
Otimização do método de localização de Monte Carlo para  
redes de sensores sem fio móveis / João Otávio Cadó de  
Matos.- 2016.  
128 p.; 30 cm

Orientador: João Baptista dos Santos Martins  
Dissertação (mestrado) - Universidade Federal de Santa  
Maria, Centro de Tecnologia, Programa de Pós-Graduação em  
Informática, RS, 2016

1. Rede de sensores sem fio 2. Algoritmo de  
localização 3. Método de Monte Carlo I. Martins, João  
Baptista dos Santos II. Título.

---

© 2016

Todos os direitos autorais reservados a João Otávio Cadó de Matos. A reprodução de partes ou do todo deste trabalho só poderá ser feita mediante a citação da fonte.

E-mail: jota.cado@gmail.com

**João Otávio Cadó de Matos**

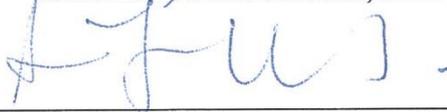
**OTIMIZAÇÃO DO MÉTODO DE LOCALIZAÇÃO DE MONTE CARLO PARA  
REDES DE SENSORES SEM FIO MÓVEIS**

Dissertação apresentada ao Curso de Mestrado do Programa de Pós-Graduação em Informática (PPGI), área de concentração em Computação, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Mestre em Informática**.

**Aprovado em 16 de setembro de 2016:**

  
\_\_\_\_\_  
**João Baptista dos Santos Martins, Dr. (UFSM)**  
(Presidente/Orientador)

  
\_\_\_\_\_  
**Roseclea Duarte Medina, Dra. (UFSM)**

  
\_\_\_\_\_  
**Sérgio José Melo de Almeida, Dr. (UCPel)**

Santa Maria, RS  
2016



## RESUMO

### OTIMIZAÇÃO DO MÉTODO DE LOCALIZAÇÃO DE MONTE CARLO PARA REDES DE SENSORES SEM FIO MÓVEIS

AUTOR: João Otávio Cadó de Matos  
ORIENTADOR: João Baptista dos Santos Martins

O crescente desenvolvimento da tecnologia de integração de circuitos eletrônicos aliado à evolução dos sistemas microeletromecânicos vem possibilitando um rápido avanço das pesquisas na área de redes de sensores sem fio, as quais são formadas por um conjunto de dispositivos sensores com recursos de processamento, memória e potência extremamente limitados. Graças a tais avanços, as redes de sensores sem fio vêm sendo empregadas em um número cada vez maior de aplicações envolvendo atividades militares e médicas, monitoramento ambiental, domiciliar e industrial, controle de tráfego, entre outras. Em muitas dessas aplicações, o conhecimento da localização dos nodos sensores é requisito fundamental, além de contribuir para o aumento da segurança e da vida útil da rede. O uso de dispositivos GPS para cumprir tal função de localização, no entanto, torna-se inviável à medida que aumenta a densidade de nodos da rede, tendo em vista o aumento do custo de implementação e do consumo de potência. Diante disso, inúmeros algoritmos são propostos com o intuito de baratear e/ou tornar mais preciso o processo de localização de nodos em redes de sensores sem fio, sejam elas móveis ou estáticas. O presente trabalho, por sua vez, apresenta um esquema de localização de nodos baseado no algoritmo de localização de Monte Carlo para redes de sensores sem fio móveis. Por meio das alterações propostas sobre tal algoritmo, objetivou-se a redução do erro de estimativa de localização, principalmente em cenários com baixa densidade de nodos portadores de dispositivo GPS. Para avaliar o comportamento do erro de estimativa do esquema proposto em função de alguns parâmetros da rede e do próprio método, desenvolveu-se uma ferramenta de simulação no *software* MatLab, a qual se mostrou bastante versátil, dados os parâmetros que permite manipular e analisar, e de fácil utilização. Dessa forma, o esquema de localização proposto foi simulado e comparado com alguns dos principais métodos de localização derivados do método de Monte Carlo para redes de sensores sem fio, adotando o erro de estimativa de localização como métrica de comparação. Considerando cenários com diferentes densidades de dispositivos GPS e com nodos movendo-se de acordo com uma versão modificada no modelo de mobilidade *random waypoint*, o esquema de localização proposto apresentou um erro de localização inferior àquele verificado para os demais métodos analisados.

**Palavras-chave:** Rede de sensores sem fio. Algoritmo de localização. Método de Monte Carlo.



## ABSTRACT

### OPTIMIZATION OF THE MONTE CARLO LOCALIZATION METHOD FOR MOBILE WIRELESS SENSOR NETWORKS

AUTHOR: João Otávio Cadó de Matos  
ADVISOR: João Baptista dos Santos Martins

The increasing development of electronic circuits integration technology combined with the evolution of the microelectromechanical systems has enabled rapid advancement of research in the area of wireless sensor networks, which are composed by a set of sensor devices with extremely limited resources in the processing power, data storage capacity and energy consumption. Thanks to these advances, wireless sensor networks are being used in an increasing number of applications involving military and medical activities, environmental, domestic and industrial monitoring, traffic control, among others. In many of these applications, the knowledge of the location of the sensor nodes is a fundamental require, besides contribute to increasing the safety and lifetime of the network. The use of GPS devices to meet such a localization function, however, becomes impractical as the network nodes density increases, in view of the increasing cost of implementation and power consumption. Thus, several algorithms are proposed to cheapen and/or improve the precision of the nodes localization process in wireless sensor networks, whether mobile or static. Therefore, this work presents a node localization scheme based on Monte Carlo localization algorithm for mobile wireless sensor networks. With the proposed amendments on such an algorithm, the main objective was to reduce the estimate location error, especially in scenarios with low GPS devices density. To evaluate the behavior of the proposed scheme estimate error against some parameters of the network and the method itself, it was developed a simulation tool in MatLab software, which has proved quite versatile, given the parameters to manipulate and analyze, and easy to use. Thus, the proposed localization scheme was simulated and compared to some of the main localization algorithms derived from Monte Carlo method for wireless sensor networks, adopting the estimate location error as the comparison metric. Considering scenarios with different GPS devices densities and with nodes moving according to a modified version of the random waypoint mobility model, the proposed localization scheme presented a location error lower than that observed for the other methods analyzed.

**Key-words:** Wireless sensor network. Localization algorithm. Monte Carlo method.



## LISTA DE FIGURAS

Figura 1 -	Organização clássica de uma RSSF. ....	17
Figura 2 -	Componentes de um nodo sensor. ....	18
Figura 3 -	Arquitetura <i>ad hoc</i> da rede desenvolvida para o projeto ZebraNet (JUAN-ANG, OKI, <i>et al.</i> , 2002). ....	20
Figura 4 -	Fluxo simplificado do funcionamento do Sistema de Posicionamento Global. ....	23
Figura 5 -	Interpretação geométrica do método de trilateração. Para a localização de um nodo em um sistema de coordenadas 2D são necessários, pelo menos, 3 nodos de referência. ....	24
Figura 6 -	Estimativa das coordenadas ( $d_x$ , $d_y$ ) de um nodo desconhecido $d$ através do método TDOA. ....	27
Figura 7 -	Triangulação no método AOA. (a) Nodo desconhecido $d$ conhece sua orientação; (b) Nodo desconhecido $d$ desconhece sua orientação. ....	29
Figura 8 -	Algoritmo CAB: dois níveis de potência são utilizados para determinar os pontos de intersecção válidos e estimar a localização de nodos desconhecidos em uma abordagem <i>range-free</i> . ....	34
Figura 9 -	Aumento da densidade de nodos âncoras mantendo o alcance de transmissão. (a) Distribuição regular de 4 nodos âncoras; (b) Distribuição regular de 9 nodos âncoras. ....	36
Figura 10 -	Localização do nodo desconhecido $D$ através do método DV-HOP. ....	38
Figura 11 -	Fonte de erro no algoritmo DV-HOP. ....	40
Figura 12 -	Estimativa de localização de um nodo desconhecido através do algoritmo DRL: etapa de triangulação robusta. (a) Situação onde há apenas dois nodos de referência; (b) Situação onde há um único nodo de referência. ....	42
Figura 13 -	<i>Backup</i> inteligente: estimativa de localização de um nodo desconhecido através do algoritmo DRL quando não há nodos de referência. ....	43
Figura 14 -	Exemplo do movimento aleatório dos nodos representados pelos animais de um rebanho bovino. Aqui, $t_0$ e $t_1$ ( $t_1 > t_0$ ) indicam instantes de tempo distintos, evidenciando a movimentação de alguns nodos da rede e o estado estacionário de outros. ....	46
Figura 15 -	Densidades de probabilidade e conjuntos de amostras para uma iteração do método de localização de Monte Carlo aplicado à robótica móvel. ....	54
Figura 16 -	Etapa de inicialização do algoritmo de localização de Monte Carlo para RSSFs móveis. Os nodos desconhecidos procuram por nodos âncoras em suas respectivas vizinhanças. Caso verifiquem a presença de pelo menos um âncora, os nodos desconhecidos geram as amostras $s_0^i$ ( $i \leq N$ ) baseando-se no fato de estarem dentro do raio de alcance do âncora de coordenadas $(x_a, y_a)$ . ....	57
Figura 17 -	Etapa de predição do algoritmo de localização de Monte Carlo para RSSFs móveis. Os três pontos representados na cor preta, situados nos centros das circunferências de raios $v_{max}$ , correspondem a amostras do conjunto $S_{k-1}$ utilizado para estimar a posição de um nodo qualquer na iteração anterior (instante de tempo $k-1$ ). Os pontos distribuídos em torno dessas três amostras pertencem ao conjunto $S'_k$ , formado a partir da aplicação do modelo de mobilidade sobre cada uma das amostras em $S_{k-1}$ .	

.....	58
Figura 18 - Classificação dos nodos âncoras: o nodo âncora da figura se movimenta da posição $l_{k-1}$ no instante de tempo $k-1$ para a posição $l_k$ no instante de tempo $k$ . Para nodos situados na região III, o âncora é classificado como <i>insider</i> . Para nodos situados na região II, o âncora é classificado como <i>arriver</i> . Para os nodos da região I, o âncora é um <i>leaver</i> . .....	59
Figura 19 - Tomando o nodo desconhecido $d$ como referência, os âncoras $a_1$ e $a_2$ são considerados âncoras diretos, uma vez que estão na vizinhança de $d$ . O âncora $a_3$ , por sua vez, é um âncora indireto para $d$ , uma vez que está na vizinhança de $n$ , mas não está na vizinhança de $d$ . Nesse caso, $d$ recebe as informações de localização do âncora $a_3$ por meio de $n$ . .....	61
Figura 20 - Influência dos buracos da rede sobre a precisão das estimativas de localização do algoritmo de Monte Carlo para RSSFs móveis. ....	64
Figura 21 - Impacto da densidade de nodos âncoras da rede sobre o erro médio de estimativa do algoritmo de localização de Monte Carlo para RSSFs. ....	67
Figura 22 - O pior caso possível para a situação onde o algoritmo de localização utiliza-se apenas do modelo de mobilidade baseado na velocidade máxima para estimar a localização de um nodo desconhecido $d$ situado em um buraco dentro da rede. No exemplo ilustrado, são consideradas 3 iterações do algoritmo. ....	69
Figura 23 - Construção da <i>anchor box</i> . ....	70
Figura 24 - Setor que contém as amostras do conjunto $S'_k$ . ....	73
Figura 25 - Contribuição negativa das amostras $s_k^{i-1}$ e $s_k^i$ ao processo de localização do algoritmo de Monte Carlo quando se considera apenas a velocidade máxima dos nodos. No caso ilustrado, $l_{k-1}^{est}$ corresponde à localização estimada do nodo $d$ no instante de tempo $k-1$ e $l_k$ corresponde à localização real do nodo $d$ no instante de tempo $k$ . ....	74
Figura 26 - Definição da área de predição do esquema de localização proposto. ....	85
Figura 27 - Áreas de predição sobre as quais atuam as funções de amostragem do método MCL original, do método OTMCL e do esquema de localização proposto. ....	86
Figura 28 - Diagrama da primeira etapa da primeira etapa de funcionamento do simulador desenvolvido. ....	94
Figura 29 - Modelo de mobilidade <i>Random Waypoint</i> modificado. ....	96
Figura 30 - Gráficos gerados pelo simulador desenvolvido mostrando a topologia da rede para cada um dos instantes de simulação definidos pelo passo de simulação. ....	98
Figura 31 - Comparação entre simuladores por meio da relação entre o erro médio de estimativa do algoritmo MCL e o tempo de simulação. (a) Gráfico obtido através da ferramenta de simulação desenvolvida no presente trabalho; (b) Gráfico obtido através do simulador disponível em (UNIVERSITY OF VIRGINIA, 2004). ....	101
Figura 32 - Comparação entre simuladores por meio da relação entre o erro de estimativa do algoritmo MCL e a densidade de nodos âncoras da rede. (a) Gráfico obtido através da ferramenta de simulação desenvolvida no presente trabalho; (b) Gráfico obtido através do simulador disponível em (UNIVERSITY OF VIRGINIA, 2004). ....	103
Figura 33 - Impacto da densidade de nodos âncoras sobre o erro médio de estimativa durante a etapa de inicialização do esquema de localização proposto	

	(ELP) e dos algoritmos MCL, MCB e OTMCL. ....	106
Figura 34 -	Precisão das estimativas do ELP em função do tempo para diferentes valores de $\varepsilon$ . ....	111
Figura 35 -	Erro médio de estimativa do ELP para diferentes valores de $\varepsilon$ . ....	112
Figura 36 -	Precisão das estimativas de localização em função do tempo de simulação para o ELP e para os métodos de localização MCL, MCB e OTMCL. ....	113
Figura 37 -	Impacto da densidade de nodos âncoras sobre o erro de estimativa do esquema de localização proposto (ELP) e dos algoritmos MCL, MCB e OTMCL. ....	114
Figura 38 -	Precisão das estimativas do ELP em função do tempo para diferentes valores de $N$ . ....	116
Figura 39 -	Erro médio de estimativa do ELP para diferentes valores de $N$ . ....	117



## LISTA DE TABELAS

Tabela 1 -	Parâmetros de simulação utilizados para avaliar a influência dos chamados buracos da rede sobre o processo de estimativa de localização dos nodos desconhecidos. ....	65
Tabela 2 -	Parâmetros utilizados para simular o modelo de mobilidade implementado. ....	96
Tabela 3 -	Parâmetros de simulação utilizados para ilustrar a primeira etapa de funcionamento da ferramenta desenvolvida. ....	97
Tabela 4 -	Parâmetros de simulação utilizados na análise comparativa entre a ferramenta implementada no software MatLab e aquela disponibilizada em (UNIVERSITY OF VIRGINIA, 2004). ....	101
Tabela 5 -	Parâmetros de simulação utilizados para avaliar a etapa de inicialização do ELP e dos métodos MCL, MCB e OTMCL. ....	105
Tabela 6 -	Redução do erro médio de estimativa durante a etapa de inicialização do ELP em comparação aos métodos MCL, MCB e OTMCL. ....	108
Tabela 7 -	Percentual de nodos desconhecidos sem estimativa de localização durante a etapa de inicialização do ELP e dos métodos MCL, MCB e OTMCL. ....	108
Tabela 8 -	Parâmetros de simulação utilizados para avaliar o impacto do parâmetro $\varepsilon$ sobre o erro médio de estimativa do ELP. ....	110



## LISTA DE ABREVIATURAS E SIGLAS

AOA	<i>Angle of Arrival</i>
CAB	<i>Concentric Anchor-Beacons</i>
DRL	<i>Dynamic Reference Localization</i>
ELP	Esquema de Localização Proposto
<i>fh</i>	<i>flooding-hop</i>
GPS	<i>Global Positioning System</i>
<i>hd</i>	<i>hop-distance</i>
IEEE	<i>Institute of Electric and Electronic Engineers</i>
MCB	<i>Monte Carlo Boxed</i>
MCL	<i>Monte Carlo Localization</i>
MEMS	<i>Microelectromechanical Systems</i>
OTMCL	<i>Oriented Tracking-based Monte Carlo Localization</i>
PDF	<i>Probability Density Function</i>
RD	<i>Random Direction</i>
RF	Radiofrequência
RPGM	<i>Reference Point Group Mobility</i>
RSSF	Rede de Sensores Sem Fio
RSSI	<i>Received Signal Strength Indicator</i>
RWP	<i>Random Waypoint</i>
SANLA	<i>Selective Anchor Node Localization Algorithm</i>
TDOA	<i>Time Difference of Arrival</i>
TOA	<i>Time of Arrival</i>
US DARPA	<i>United States Defense Advanced Research Projects Agency</i>



## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	12
1.1	MOTIVAÇÃO .....	13
1.2	CONTRIBUIÇÕES .....	15
1.3	ORGANIZAÇÃO DO TRABALHO.....	15
<b>2</b>	<b>LOCALIZAÇÃO EM RSSFs</b> .....	16
2.1	RSSF .....	16
2.1.1	<b>Redes <i>ad hoc</i></b> .....	19
2.2	ALGORITMOS DE LOCALIZAÇÃO EM RSSFs.....	20
2.2.1	<b>O Sistema de Posicionamento Global (GPS)</b> .....	22
2.2.2	<b>Métodos <i>range-based</i></b> .....	25
2.2.3	<b>Métodos <i>range-free</i></b> .....	32
2.2.4	<b>O impacto da mobilidade sobre algoritmos de localização em RSSFs</b> .....	45
2.3	RESUMO DO CAPÍTULO .....	48
<b>3</b>	<b>MÉTODO DE MONTE CARLO</b> .....	50
3.1	INFERÊNCIA BAYESIANA.....	50
3.2	MÉTODO DE LOCALIZAÇÃO DE MONTE CARLO .....	53
3.3	ALGORITMO DE LOCALIZAÇÃO DE MONTE CARLO PARA RSSFs MÓVEIS.....	55
3.3.1	<b>Limitações do algoritmo de localização de Monte Carlo para RSSFs móveis</b> .....	62
3.3.2	<b>Trabalhos relacionados</b> .....	69
3.4	RESUMO DO CAPÍTULO .....	75
<b>4</b>	<b>ESQUEMA DE LOCALIZAÇÃO PROPOSTO</b> .....	78
4.1	ETAPA DE INICIALIZAÇÃO .....	78
4.2	ETAPAS DE PREDIÇÃO E DE FILTRAGEM .....	83
4.3	RESUMO DO CAPÍTULO .....	90
<b>5</b>	<b>FERRAMENTA DE VALIDAÇÃO DOS RESULTADOS</b> .....	92
5.1	FUNCIONAMENTO DA FERRAMENTA DESENVOLVIDA.....	93
5.1.1	<b>Construção da topologia da rede</b> .....	93
5.1.2	<b>Simulação do algoritmo de localização</b> .....	99
5.2	RESUMO DO CAPÍTULO .....	103
<b>6</b>	<b>ANÁLISE DOS RESULTADOS</b> .....	104
6.1	RESULTADOS OBTIDOS COM A ETAPA DE INICIALIZAÇÃO.....	105
6.2	RESULTADOS OBTIDOS COM O ESQUEMA DE LOCALIZAÇÃO PROPOSTO.....	109
6.2.1	<b>O impacto do parâmetro <math>\varepsilon</math> sobre o erro de estimativa</b> .....	110
6.2.2	<b>Precisão das estimativas de localização em função do tempo</b> .....	112
6.2.3	<b>O impacto da densidade de nodos âncoras sobre o erro de estimativa</b> .....	114
6.2.4	<b>O impacto da quantidade de amostras sobre o erro de estimativa</b> .....	116
6.3	RESUMO DO CAPÍTULO .....	117
<b>7</b>	<b>CONCLUSÃO</b> .....	120
	<b>REFERÊNCIAS</b> .....	122



## 1 INTRODUÇÃO

O ato de monitorar eventos e/ou processos passa a ser uma atividade vital para a humanidade a partir do momento em que o ser humano percebe que a observação de determinados fenômenos revela informações úteis à tomada de decisões. Em outras palavras, ao prestar atenção, por exemplo, ao comportamento das condições climáticas de determinada região ao longo do ano, o homem foi capaz de determinar as épocas propícias à pesca, à caça e ao cultivo de diferentes espécies vegetais. Nesse momento histórico, obviamente, a tecnologia disponível para a realização de tais observações agregava pouco ou nenhum conhecimento científico, e, da mesma forma, os processos a serem observados eram, para os dias de hoje, extremamente simples.

No entanto, a cada passo de desenvolvimento da tecnologia, a humanidade depara-se com novos dilemas, sendo essa a relação que faz avançar o conhecimento científico e, conseqüentemente, o desenvolvimento tecnológico. Assim, com o passar do tempo, desenvolve-se a necessidade de monitorar eventos e/ou processos cada vez mais complexos, de forma a obter informações ainda mais precisas, específicas e relevantes.

Nesse contexto, a miniaturização<sup>1</sup> aplicada ao campo da eletrônica desempenha um papel determinante no processo de desenvolvimento das chamadas Redes de Sensores Sem Fio (RSSFs). O rápido avanço na tecnologia de integração de circuitos eletrônicos em silício (MACK, 2011) permitiu o desenvolvimento de sistemas embarcados com capacidades de processamento e de comunicação sem fio surpreendentes. Aliada a isso, a crescente evolução dos sistemas microeletromecânicos (MEMS, do inglês *Microelectromechanical Systems*) (GARDNER e AWADELKARIM, 2001) possibilitou o surgimento dos chamados nodos, dispositivos formados, normalmente, por sensores, microcontrolador, comunicação por radiofrequência (RF) e bateria.

Uma RSSF consiste, basicamente, em uma rede de nodos sensores organizados de forma a cooperarem entre si, objetivando monitorar eventos de interesse. Incluem-se, nesse caso, aplicações envolvendo atividades militares, monitoramento domiciliar e ambiental, sensoriamento industrial, controle de tráfego, entre outras. Tendo em vista a gama de aplicações onde pode ser empregada, em (21 ideas for the 21st century, 1999), a tecnologia de RSSFs foi considerada uma das 21 mais importantes do século XXI.

---

<sup>1</sup> Miniaturização é o nome que se dá ao processo de produção de bens de consumo cada vez menores, iniciado durante a revolução tecnológica do século XX e atualmente em andamento. A eletrônica é, de longe, o campo onde a miniaturização é mais intensa, apesar de essa ser uma tendência em todos os segmentos da produção industrial.

Em muitas dessas aplicações, é importante observar que o conhecimento acerca das localizações dos nodos dentro da área da rede mostra-se de extrema importância, pois, tendo em vista os objetivos de cada uma das aplicações, as coordenadas dos dispositivos acabam sendo determinantes na interpretação dos dados coletados. Nesse sentido, segundo (RABAEY, AMMER, *et al.*, 2000), dados coletados do meio tornam-se irrelevantes diante da impossibilidade de obtenção das coordenadas do fenômeno sob estudo. Como exemplo, toma-se a RSSF apresentada em (GREENORBS, 2011), responsável pela coleta de dados de uma floresta, incluindo temperatura, umidade, iluminação e concentração de dióxido de carbono. Nesse caso, a tomada de decisões com base nas informações coletadas pelos nodos, como o controle de um foco de incêndio, por exemplo, só é possível a partir do conhecimento da localização de cada nodo. Caso contrário, a informação coletada é inútil. Há ainda cenários onde os nodos que formam a rede são móveis, como é o caso da rede descrita em (JUANG, OKI, *et al.*, 2002), cujo objetivo é o monitoramento de um rebanho de zebras (nodos móveis) a fim de coletar dados relevantes a pesquisas sobre essa espécie animal. Nessa aplicação, é indispensável o rastreamento das trajetórias percorridas por cada animal do rebanho, visto que tal informação pode evidenciar, por exemplo, zonas que proveem alimento ao rebanho, zonas evitadas pelos animais, a existência de predadores e, até mesmo, a ocorrência de mortes dentro do rebanho.

Diante disso, inúmeras são as pesquisas voltadas aos algoritmos capazes de determinar a localização dos nodos em uma RSSF, normalmente trabalhando aspectos como o consumo de energia, o custo e a facilidade de implementação, a precisão das estimativas, a complexidade computacional e o desenvolvimento de ferramentas de validação.

## 1.1 MOTIVAÇÃO

Os crescentes avanços científicos na área de RSSFs e o conseqüente aumento das aplicações envolvendo essa tecnologia acabam impondo novos desafios e novas limitações às pesquisas na área. Segundo (CHONG e KUMAR, 2003), o desenvolvimento de RSSFs depende de três áreas de pesquisas distintas: sensoriamento, comunicação e computação (*hardware*, *software* e algoritmos). Entretanto, é de fundamental importância ressaltar que as unidades de sensoriamento, comunicação e computação de um nodo sempre devem operar dentro dos limites de consumo impostos pela fonte de alimentação do dispositivo.

Na maioria das aplicações em RSSFs móveis, os nodos são alimentados por baterias e, dependendo da aplicação, devem operar por um longo período de tempo de forma autônoma,

sem receber qualquer tipo de manutenção ou reposição de componentes (AKYILDIZ, SU, *et al.*, 2002). Em (JUANG, OKI, *et al.*, 2002), por exemplo, a rede deve operar pelo período mínimo de um ano sem qualquer intervenção humana direta. Por isso, o consumo de energia por parte dos nodos representa uma grande limitação no projeto dessa rede.

No que diz respeito aos algoritmos de localização e/ou rastreamento desenvolvidos sobre RSSFs, o consumo de potência está diretamente associado à densidade de nodos âncoras<sup>2</sup> necessária para a obtenção das estimativas de localização. Isso acontece pelo fato de esses nodos normalmente adquirirem suas coordenadas por meio de receptores GPS, o que acaba gerando, também, um aumento nos custos econômicos de implementação da rede. Em função disso, muitas das pesquisas envolvendo algoritmos de localização e de rastreamento em redes de sensores focam na redução do número de nodos âncoras da rede sem a perda (se possível, com aumento) de precisão nas estimativas de localização, como é o caso dos algoritmos apresentados em (MARTINS, CHEN e SEZAKI, 2009) e (HSIEH e WANG, 2006).

Além das limitações energéticas, comuns a redes estáticas e móveis, o aspecto da mobilidade da rede também impõe grandes desafios ao processo de localização de nodos em RSSFs móveis. Nesse caso, a lógica de funcionamento do algoritmo deve ser pensada para diferentes configurações topológicas da rede em estudo, considerando diferentes modelos de mobilidade para os nodos. Ao contrário de uma rede estática, onde, idealmente, a descoberta das posições dos nodos é realizada apenas uma vez, redes móveis exigem que a computação do método de localização seja recorrente. Esse fator, portanto, também acaba influenciando no consumo de energia por parte dos nodos. Assim, as pesquisas na área buscam constantemente a redução da complexidade computacional dos métodos de localização utilizados.

Diante da vasta quantidade de aplicações e desses inúmeros fatores que influenciam no projeto e na implementação das RSSFs e dos algoritmos nelas empregados, a utilização de simuladores para a validação de resultados torna-se cada vez mais indispensável. Segundo (LEVIS, LEE, *et al.*, 2003), os simuladores cumprem um importante papel ao possibilitarem avaliar alternativas de projeto, explorar configurações difíceis de serem implementadas na prática e observar interações que também se mostram difíceis de serem capturadas em um sistema real. No entanto, para o caso específico da validação de algoritmos de localização em RSSFs móveis, há determinadas métricas, medidas e parâmetros de entrada para os quais os simuladores frequentemente utilizados na literatura (NS-2 (MOZUMDAR, GREGORETTI, *et*

---

<sup>2</sup> Nodos que possuem conhecimento acerca de suas coordenadas dentro da rede. Por isso, são tomados como referências para que os demais nodos da rede estimem suas localizações. Normalmente, a informação de localização é fornecida aos nodos âncoras por meio de receptores GPS. Há casos, normalmente em redes estáticas, em que a localização é informada manualmente.

*al.*, 2008), TOSSIM (ZHENYU, 2009) e OMNeT++ (STEHLIK, 2011)) não oferecem suporte e/ou flexibilidade.

Assim, o presente trabalho propõe um método para a localização de nodos em RSSFs móveis e também uma ferramenta de simulação para esse tipo de rede, a qual é utilizada para validar os resultados do algoritmo proposto.

## 1.2 CONTRIBUIÇÕES

Dentre as contribuições do presente trabalho, cita-se o desenvolvimento de um método *range-free*<sup>3</sup> para localização de nodos em RSSFs móveis baseado em algoritmos de baixa complexidade computacional encontrados na literatura científica da área. A técnica apresentada propõe soluções para a redução da densidade de nodos âncoras e aumento na precisão das estimativas geradas a partir do método de localização de Monte Carlo para RSSFs (HU e EVANS, 2004). Além disso, o trabalho apresenta uma ferramenta com suporte à simulação de RSSFs móveis desenvolvida no *software* MatLab (MATHWORKS, 2012) e utilizada para validar os resultados do esquema de localização proposto.

## 1.3 ORGANIZAÇÃO DO TRABALHO

A presente dissertação encontra-se organizada em sete capítulos. O Capítulo 2 apresenta uma revisão teórica acerca dos principais conceitos relacionados a RSSFs e métodos de localização de nodos, introduzindo os principais algoritmos encontrados na literatura científica da área. O Capítulo 3 apresenta uma análise minuciosa do método de localização de Monte Carlo bem como de alguns dos principais algoritmos de localização para RSSFs derivados de tal método. Em seguida, no Capítulo 4, detalha-se o desenvolvimento do esquema de localização *range-free* proposto. No Capítulo 5, descrevem-se a implementação da ferramenta de simulação desenvolvida e sua validação através da simulação do algoritmo de localização de Monte Carlo para RSSFs móveis. O Capítulo 6 promove uma discussão acerca dos resultados obtidos com o esquema de localização proposto. Por fim, no sétimo e último capítulo, são pontuadas as principais contribuições desta dissertação e algumas sugestões para trabalhos futuros.

---

<sup>3</sup> Métodos de localização que se baseiam apenas na conectividade entre os nodos para estimar as coordenadas dos mesmos.

## 2 LOCALIZAÇÃO EM RSSFs

As RSSFs estão cada vez mais presentes em nosso cotidiano, sendo utilizadas em uma vasta gama de aplicações, e, em muitas dessas aplicações, torna-se necessário o conhecimento acerca das posições e/ou, para o caso específico de redes formadas por sensores móveis, das trajetórias percorridas pelos nodos que formam a rede. Monitoramento ambiental (DARDARI, CONTI, *et al.*, 2007) e animal (JUANG, OKI, *et al.*, 2002), aplicações militares e controle de tráfego são exemplos de aplicações de RSSFs que necessitam de informações a respeito da localização dos nodos para o seu funcionamento. Em (JUANG, OKI, *et al.*, 2002), por exemplo, é apresentado o projeto denominado ZebraNet, que consiste em uma RSSF responsável pelo monitoramento das zebras da região de Laikipia, no Quênia. Já em (SATHYAN e HEDLEY, 2013), os autores apresentam como cenário motivacional para a aplicação do algoritmo de rastreamento proposto, uma RSSF formada por dispositivos conectados ao corpo de bombeiros em atuação dentro de um prédio. Nesse caso específico, o conhecimento acerca da localização dos bombeiros no interior do prédio permite, caso seja necessária, uma evacuação planejada e eficaz. Em ambos os exemplos citados, a determinação da localização dos nodos dentro da rede é uma etapa essencial para o correto funcionamento da aplicação proposta. Além disso, conforme (BACHIR, DOHLER e WATTEYNE, 2010) e (HU e EVANS, 2004), as informações sobre a localização dos nodos também podem auxiliar na economia de energia, no roteamento de pacotes e na segurança das informações que transitam na rede.

Diante disso, a ideia de um algoritmo de localização em uma RSSF é situar, dentro da área da rede, as coordenadas, sejam elas relativas ou absolutas, dos nodos que integram essa rede (OLIVEIRA, 2009).

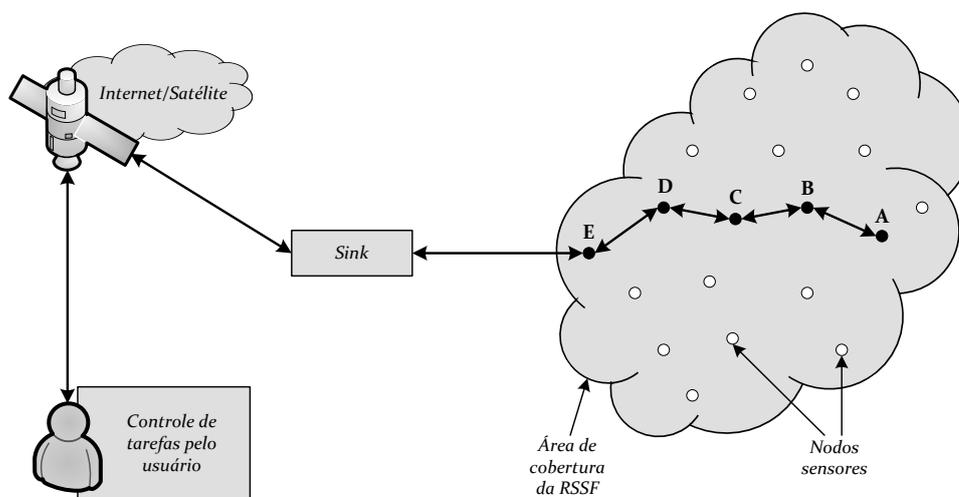
O presente capítulo apresenta uma breve introdução acerca da teoria relacionada a RSSFs. Em seguida, são apresentados alguns dos algoritmos de localização encontrados no meio científico, focando naqueles que se aplicam a redes móveis de sensores, sobretudo, quando não há a influência de atuadores na movimentação.

### 2.1 RSSF

Uma RSSF consiste, basicamente, em uma rede de nodos sensores, alguns deles especializados ou não, organizados de tal forma a cooperarem entre si, objetivando monitorar determinados eventos a fim de coletar informações do ambiente. Incluem-se, nesse caso, tarefas como vigilância, rastreamento, coleta de dados, entre outras. A Figura 1 apresenta uma orga-

nização clássica para esse tipo de rede, na qual as informações são coletadas pelos nodos e roteadas através dos mesmos até o *sink*, um nodo responsável por agrupar essas informações e transmiti-las para um centro de controle. Como exemplo de uma RSSF real, pode-se citar a GreenOrbs (GREENORBS, 2011), uma rede responsável pela coleta de dados de uma floresta, incluindo temperatura, umidade, iluminação e concentração de dióxido de carbono.

Figura 1 - Organização clássica de uma RSSF.

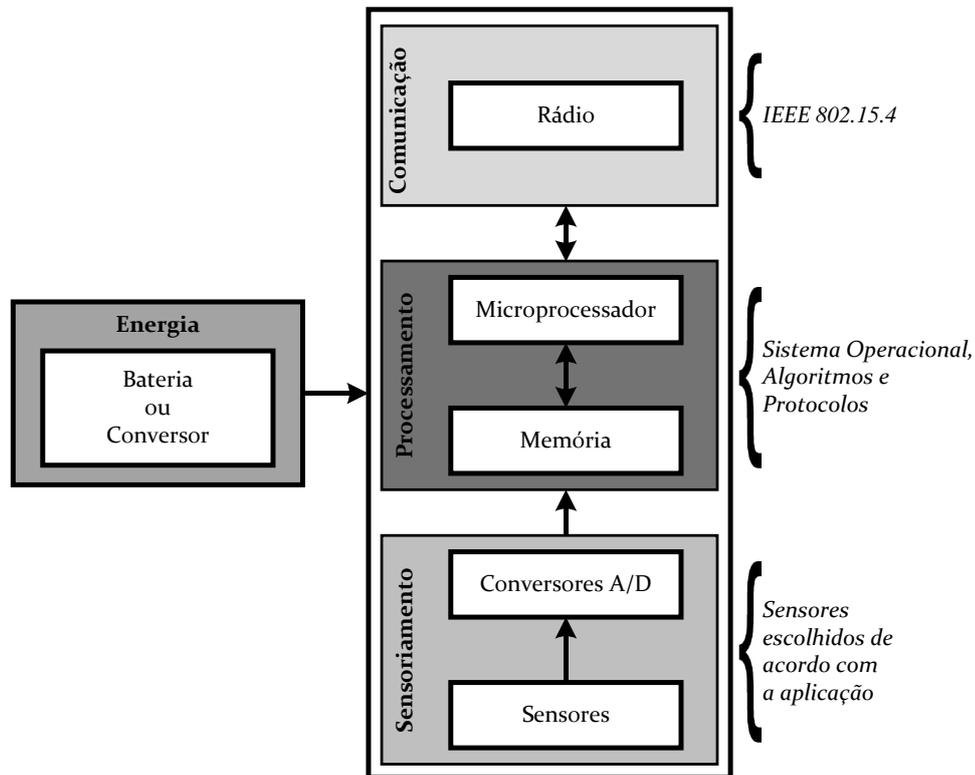


Fonte: Adaptação de (AKYILDIZ, SU, *et al.*, 2002).

A partir dos anos 2000, as pesquisas na área de RSSFs foram potencializadas pelo crescente avanço nas áreas de comunicação sem fio, eletrônica digital e sistemas microeletromecânicos (MEMS, do inglês *Microelectromechanical Systems*). Tais avanços permitiram o desenvolvimento de nodos sensores multifuncionais, de tamanho reduzido, baixo custo e com baixo consumo de energia (AKYILDIZ, SU, *et al.*, 2002), capazes de comunicarem-se em curtas distâncias sem a necessidade de uma conexão física. Em uma RSSF, esses nodos desempenham as funções de sensoriamento, processamento de dados e comunicação, sendo, em geral, densamente distribuídos no ambiente monitorado (AHMED e AVARITSIOTIS, 2014), sobretudo em função dos limites de alcance dos rádios. Os nodos ditos especializados são aqueles que podem possuir capacidades adicionais frente aos demais, sejam elas referentes a um dispositivo de rádio com maior alcance, maior poder de processamento, sensores específicos, um sistema de autolocalização ou mesmo por uma tarefa específica dentro da rede (como é o caso do *sink*). Em geral, esse tipo de elemento em uma RSSF é pequeno em relação aos demais nodos. Isso ocorre devido a questões de custo de implementação, manu-

tenção ou fator de forma final do *hardware* (OLIVEIRA, 2009) (AKYILDIZ, SU, *et al.*, 2002).

Figura 2 - Componentes de um nodo sensor.



Fonte: Adaptação de (MULLER, 2009).

Os nodos sensores que formam uma RSSF são, geralmente, compostos pelas unidades apresentadas na Figura 2. Nesse caso, os sinais analógicos gerados pelos sensores a partir do evento observado são convertidos por um conversor analógico-digital e então submetidos à unidade de processamento, formada por um componente de armazenamento e um microprocessador. À unidade de processamento cabe a interpretação dos sinais digitais recebidos, realizando as tarefas pertinentes e armazenando os resultados obtidos. Conforme mostrado no diagrama da Figura 2, a unidade de processamento também é a responsável por manter o sistema operacional, os algoritmos e os protocolos que sustentam a aplicação. A unidade de comunicação é a responsável por conectar o nodo sensor à rede, disponibilizando aos demais nodos as informações obtidas através dos sensores ou o resultado do processamento das mesmas, e recebendo dados de nodos vizinhos, a fim de utilizá-los em sua computação local ou apenas encaminhá-los ao destinatário alvo. A unidade de alimentação, por sua vez, é responsável por suprir a energia necessária ao funcionamento das demais unidades que formam o

nodo sensor. Em uma RSSF onde os nodos não são estáticos, a alimentação dos mesmos pode ser obtida, por exemplo, a partir de uma bateria. Nesse caso, considerando o limitado tempo de vida útil desses dispositivos (PANTAZIS e VERGADOS, 2007) e, em muitos casos, a impossibilidade de substituição ou manutenção dos mesmos, como em (JUANG, OKI, *et al.*, 2002), a unidade de alimentação constitui um dos componentes mais importantes do nodo sensor. As unidades de sensoriamento, processamento e comunicação do nodo sempre operam dentro das restrições de potência impostas pela unidade de alimentação. Como alternativa ao uso de baterias, a alimentação dos nodos é, muitas vezes, obtida a partir do meio onde o nodo se encontra. A energia solar fotovoltaica (BHUVANESWARI, BALAKUMAR e VAIDEHI, 2009), a indução eletromagnética (FRIEDMAN, HEINRICH e DUAN, 1997) e a conversão eletromecânica de energia (WILLIAMS e YATES, 1995) são exemplos de fontes de energia empregadas na alimentação de nodos sensores.

### 2.1.1 Redes ad hoc

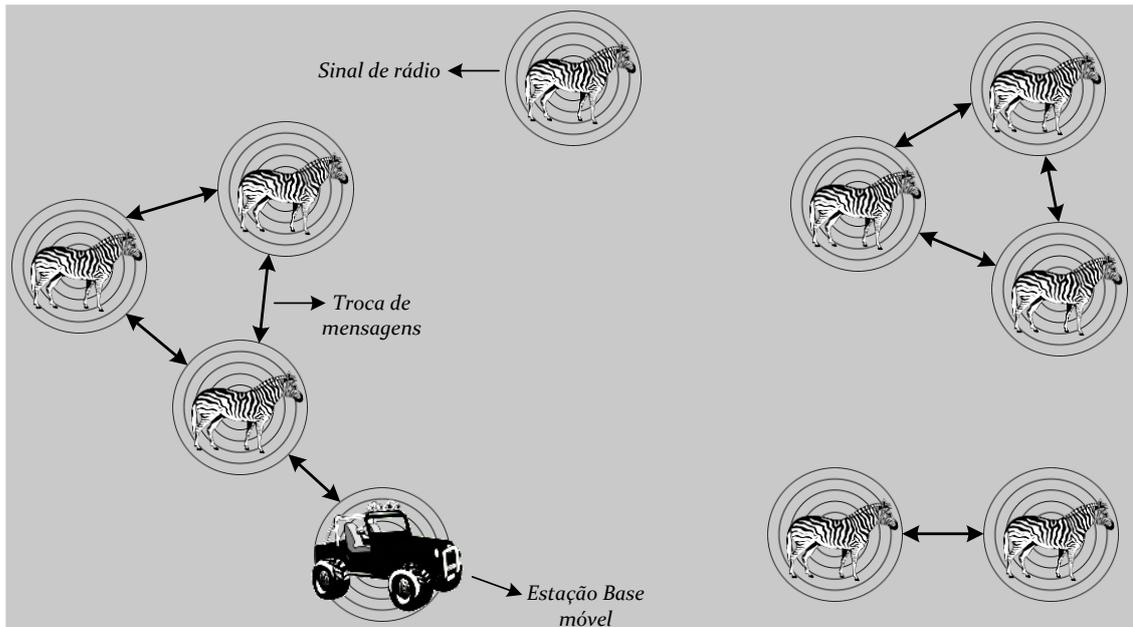
Diferente da organização clássica apresentada na Figura 1, há topologias de RSSFs que dispensam a existência de um ponto de acesso central. Nessas redes, conhecidas como redes *ad hoc*, cada nodo consegue se comunicar diretamente com os demais sem que seja necessária a criação de uma infraestrutura de rede. Nesse caso, cada dispositivo da rede atua como se fosse um roteador, operando de maneira colaborativa e repassando as informações recebidas de nodos vizinhos.

Os estudos em torno das chamadas redes *ad hoc* começam no princípio da década de 1970, quando a US DARPA (do inglês *United States Defense Advanced Research Projects Agency*) deu início a um projeto que objetivava explorar o uso de redes na comunicação via rádio em um ambiente tático militar. Uma vez que nesse tipo de rede os nodos podem se comunicar diretamente entre si, sem a necessidade de um ponto de acesso central, a arquitetura *ad hoc* acaba por aumentar de maneira significativa a flexibilidade e a mobilidade da rede. Em função disso, as MANETS (do inglês *Mobile Ad Hoc Networks*), como são chamadas as redes *ad hoc* sem fio, vêm sendo objetos de pesquisas não apenas na área de RSSFs, mas também no campo da telefonia celular e da computação.

Como exemplo de uma MANET real, pode-se citar o projeto ZebraNet (JUANG, OKI, *et al.*, 2002), uma RSSF responsável por monitorar as zebras da região de Laikipia, no Quênia. A Figura 3 apresenta a organização dessa MANET. Em função das limitações de alcance dos rádios que integram os sensores e pelo fato de a estação base ser móvel e estar disponível

esporadicamente dentro da rede, todos os nodos da rede transmitem e recebem dados dentro do alcance de seus rádios. Assim, a estação base não necessita estar dentro do alcance de determinado nodo para receber informações do mesmo.

Figura 3 - Arquitetura *ad hoc* da rede desenvolvida para o projeto ZebraNet (JUANG, OKI, *et al.*, 2002).



Fonte: Autor.

Uma vez que não existe uma hierarquia na arquitetura *ad hoc*, ou seja, todos os nodos monitoram e são monitorados, não havendo um ponto de acesso central, o roteamento das mensagens dentro da rede passa a ser uma preocupação central nessa topologia, de tal forma que a cooperação entre os nodos é de extrema importância na troca de informações. Assim sendo, todos os nodos possuem o mesmo grau de importância para fazer com que as mensagens cheguem até o seu destino final, ainda mais em redes móveis, onde a mobilidade dificulta a criação de rotas (GRACIOLI e NUNES, 2006).

## 2.2 ALGORITMOS DE LOCALIZAÇÃO EM RSSFs

Dentre os métodos utilizados para determinar a localização de nodos dentro de uma rede de sensores, o Sistema de Posicionamento Global (GPS, do inglês *Global Positioning System*) é o mais convencional (WELLENHOFF, LICHTENEGGER e COLLINS, 1997). No entanto, de acordo com (ZHAO, XI, *et al.*, 2013), essa técnica apresenta falhas ao ser utilizada

em regiões encobertas como edificações, ambientes subterrâneos e matas densas. Além disso, a utilização de GPS implica em alto consumo de energia por parte dos nodos e elevação nos custos de implementação da rede, o que, para a maioria das RSSFs, torna-se inviável (LADD, BEKRES, *et al.*, 2005) (BULUSU, HEIDEMANN e ESTRIN, 2000).

Diante dessas considerações, algumas técnicas de localização que exploram a não utilização ou o uso reduzido do número de dispositivos GPS dentro da rede são propostas como alternativas. Tais métodos ou algoritmos baseiam-se na troca de informações entre nodos âncoras (nodos que conhecem suas coordenadas, seja pelo uso de GPS ou por configuração prévia) e nodos que desconhecem suas coordenadas dentro da rede (ao longo do presente documento, estes nodos serão tratados como nodos desconhecidos). Levando essa abordagem em consideração, os algoritmos de localização são classificados em duas categorias principais, conforme (OLIVEIRA, 2009): *range-based*, baseada na medição das distâncias entre os nodos, e *range-free*, baseada apenas na conectividade entre os nodos. Ambas as técnicas serão exploradas em maiores detalhes no decorrer do presente capítulo.

A análise de algoritmos de localização aplicados a RSSFs, por sua vez, exige que determinados fatores sejam levados em consideração, sobretudo quando a rede inclui dispositivos móveis em sua topologia. Segundo (OLIVEIRA, 2009), dentre esses fatores, é preciso avaliar, por exemplo:

- como os nodos móveis da rede adquirem movimento;
- se algum dos nodos móveis necessita de equipamento mecânico adicional;
- qual o modelo de mobilidade utilizado durante as simulações;
- em que plataforma foram validados os resultados.

Todos esses pontos impactam em diversos aspectos relacionados ao desempenho do algoritmo, como o consumo de potência por parte dos nodos, a escalabilidade da RSSF, a medida da convergência dos resultados e o compromisso com a aplicação alvo. Essas características irão trilhar a especificidade e, portanto, o projeto do algoritmo de localização para operar em um cenário específico (OLIVEIRA, 2009).

No que diz respeito ao último ponto mencionado, referente à validação dos resultados, (OLIVEIRA, 2009) afirma que algoritmos de localização podem ser avaliados através de:

- experimentos físicos;
- métodos analíticos;
- simuladores voltados a redes de sensores como o TOSSIM (ZHENYU, 2009) e o NS-2 (MOZUMDAR, GREGORETTI, *et al.*, 2008);

- simuladores desenvolvidos para um propósito específico como em (HU e EVANS, 2004);
- ou mesmo uma ferramenta amplamente empregada nos meios acadêmico e profissional como o *software* MatLab (MOZUMDAR, GREGORETTI, *et al.*, 2008).

Diante de todas essas especificidades citadas, conclui-se que a simples coleta de informações provenientes de artigos científicos pode se mostrar ineficiente para a realização de um estudo comparativo eficaz entre algoritmos de localização aplicados a RSSFs. Assim, considerando os objetivos do presente trabalho, procura-se, nas seções que seguem, analisar de maneira qualitativa os métodos de localização apresentados, focando em aspectos como:

- mobilidade;
- tipo de comunicação empregada (*range-based* ou *range-free*);
- necessidade de utilização de *hardware* adicional;
- necessidade de utilização de um sistema de autolocalização.

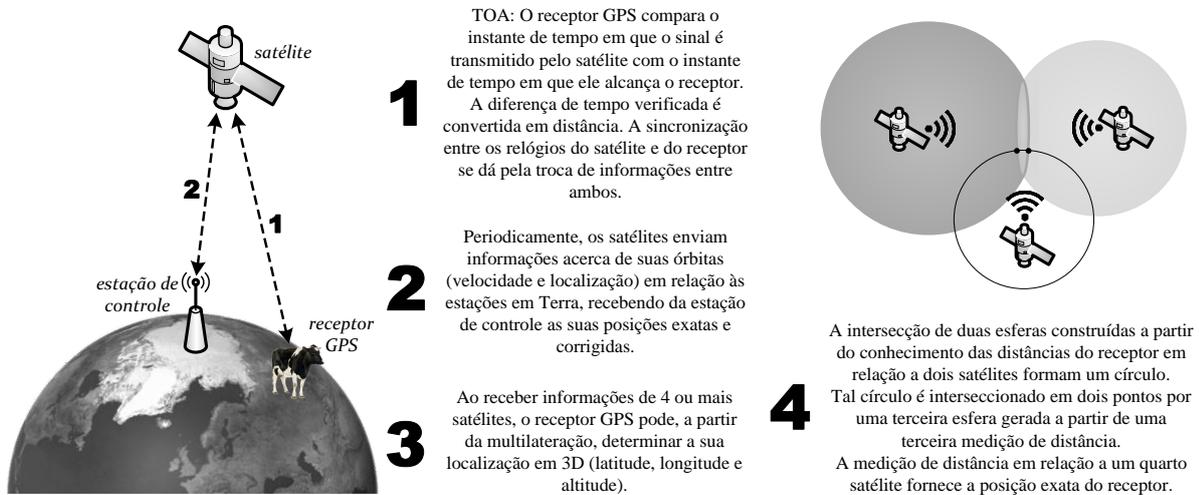
### 2.2.1 O Sistema de Posicionamento Global (GPS)

O Sistema de Posicionamento Global (GPS, do inglês *Global Positioning System*) (PARKINSON e GILBERT, 1983) (GETTING, 1993) consiste em um sistema de localização baseado em satélites que se utiliza da trilateração para determinar posições, conforme ilustra o esquema simplificado da Figura 4. O receptor GPS, acoplado ao dispositivo cuja posição deseja-se determinar, recebe sinais provenientes de pelo menos quatro satélites e, a partir da comparação entre o instante de tempo em que a mensagem é transmitida pelo satélite e o instante de tempo em que ela chega ao receptor (princípio do tempo de chegada), é possível calcular, através da Equação 1, o quão longe o receptor se encontra do satélite.

$$distancia = velocidade \times tempo \text{ de viagem} \quad (1)$$

Na equação acima, o termo *velocidade* refere-se à velocidade de propagação de uma onda de rádio no espaço, equivalente a 300.000km/s. Uma vez que a informação relacionada ao tempo de propagação do sinal é determinante para a obtenção das distâncias, é imprescindível que o sistema faça uso de relógios de alta precisão, caso contrário os erros de cálculo podem ser inaceitáveis (OLIVEIRA, 2009). Em posse das distâncias em relação aos satélites, o receptor pode, então, através da trilateração, determinar sua posição exata em um sistema de coordenadas 3D (latitude, longitude e altitude).

Figura 4 - Fluxo simplificado do funcionamento do Sistema de Posicionamento Global.



Fonte: Adaptação de (OLIVEIRA, 2009).

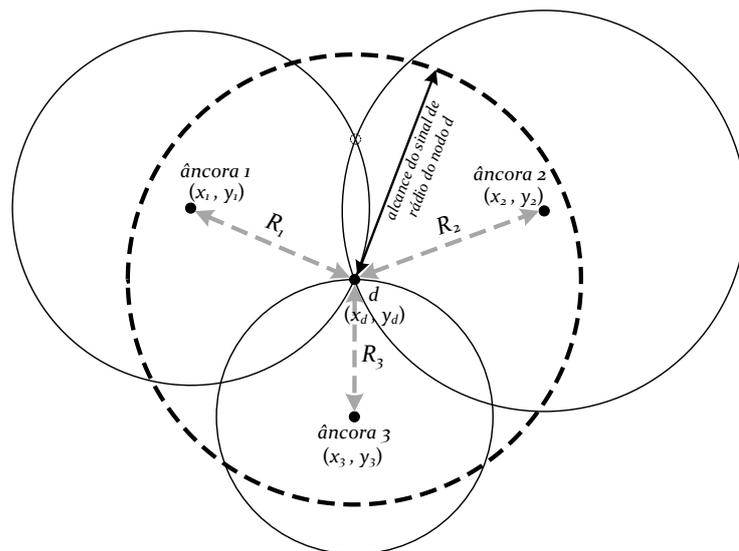
Inicialmente, o GPS foi desenvolvido com fins militares, mas encontra-se disponível para uso civil desde os anos 80. Atualmente, o sistema é composto por 24 satélites artificiais que orbitam em torno da Terra a uma distância aproximada de 19.000km. As órbitas dos satélites são definidas de tal forma que qualquer receptor GPS posicionado na superfície da Terra consiga, a qualquer momento, receber os sinais emitidos por, pelo menos, quatro satélites.

Diante das informações apresentadas e considerando o fato de o GPS estar disponível 24 horas por dia, provendo localizações absolutas sem qualquer custo financeiro de contratação do serviço, pode-se inferir que o problema de localização de nodos em redes de sensores está resolvido. Conforme já mencionado no texto introdutório do presente capítulo, apesar de o GPS ser, de fato, um sistema suficiente e eficaz para determinar a localização dos nodos de uma rede, os dispositivos são caros para serem empregados em redes muito densas, aumentam consideravelmente o consumo de energia dos nodos e, além disso, a recepção do sinal em ambientes encobertos não é garantida (LADD, BEKRES, *et al.*, 2005) (BULUSU, HEIDEMANN e ESTRIN, 2000). Apesar disso, a maioria dos algoritmos de localização e de rastreamento aplicados a redes de sensores consideram a utilização de alguns nodos com dispositivo GPS integrado (HU e EVANS, 2004) (NICULESCU e NATH, 2001). Tais nodos, referenciados como âncoras (*anchors*), *landmarks*, *beacons* ou *seeds*, pelo fato de terem informações precisas acerca de sua localização, servem como nodos de referência para que os nodos desconhecidos estimem suas coordenadas na rede.

### 2.2.1.1 Trilateração

Conforme mencionado anteriormente, o sistema GPS utiliza-se das distâncias calculadas entre o receptor e os satélites para realizar a trilateração e determinar sua posição. Tomando como exemplo o sistema de coordenadas de duas dimensões (2D) ilustrado na Figura 5, a determinação da localização do nodo desconhecido  $d$  através do método de trilateração só é possível com a obtenção de, pelo menos, três distâncias entre o nodo  $d$  e nodos de referência (em um sistema de três dimensões (3D) é necessário que haja, no mínimo, quatro pontos de referência).

Figura 5 - Interpretação geométrica do método de trilateração. Para a localização de um nodo em um sistema de coordenadas 2D são necessários, pelos menos, 3 nodos de referência.



Fonte: Adaptação de (OLIVEIRA, 2009).

Nesse caso, obtidas as distâncias  $R_1$  e  $R_2$ , o nodo desconhecido  $d$  posiciona-se em um dos pontos de intersecção entre as circunferências centradas em  $(x_1, y_1)$  e  $(x_2, y_2)$ . Uma vez que curvas quadráticas podem apresentar múltiplas intersecções no plano, uma terceira referência é necessária para determinar as coordenadas  $d_x$  e  $d_y$  do nodo  $d$  através do sistema de equações descrito pela Equação 2:

$$\begin{aligned}
 (x_1 - d_x)^2 + (y_1 - d_y)^2 &= R_1^2 \\
 (x_2 - d_x)^2 + (y_2 - d_y)^2 &= R_2^2 \\
 (x_3 - d_x)^2 + (y_3 - d_y)^2 &= R_3^2
 \end{aligned}
 \tag{2}$$

onde  $R_i$  é a distância calculada entre o nodo desconhecido e o nodo de referência  $i$ ,  $x_i$  e  $y_i$  são as coordenadas do nodo de referência  $i$  e  $d_x$  e  $d_y$  são as coordenadas do nodo  $d$ . O sistema de equações descrito pode ser resolvido, após linearizações, através do algoritmo dos mínimos quadrados.

### 2.2.2 Métodos *range-based*

Nos algoritmos *range-based*, as distâncias entre os nodos âncoras e os nodos desconhecidos são obtidas por medidas de alcance, ou *range*, do sinal de rádio. Tais medições podem ser feitas através da indicação de intensidade do sinal recebido (RSSI, do inglês *Received Signal Strength Indicator*) (BAHL e PADMANABHAN, 2000), do tempo de chegada (TOA, do inglês *Time of Arrival*) (WELLENHOFF, LICHTENEGGER e COLLINS, 1997), da diferença do tempo de chegada (TDOA, do inglês *Time Difference of Arrival*) (GUSTAFSSON e GUNNARSSON, 2003) ou do ângulo de chegada do sinal (AOA, do inglês *Angle of Arrival*) (RONG e SICHITIU, 2006). A partir das distâncias medidas, as coordenadas dos nodos desconhecidos são computadas através de métodos como a trilateração/multilateração (MULLER, 2009) e a triangulação (RONG e SICHITIU, 2006). A aplicação de tais técnicas de medição de distância, entretanto, pode requerer um hardware específico que normalmente não está integrado aos nodos sensores. O acréscimo desse hardware implica em aumento no custo e no tamanho dos nodos. Além disso, muitas das técnicas *range-based* requerem uma topologia de rede específica, com um número significativo de nodos âncoras distribuídos de tal forma que consigam cobrir toda a área da rede. Essa configuração, além de representar uma elevação nos custos, é, muitas vezes, impossível de ser implementada (como exemplo, tomemos aplicações militares onde os nodos são lançados de um avião sobre um território hostil).

A seguir, são introduzidos os principais métodos de localização *range-based* encontrados na literatura científica, apresentando a teoria relacionada ao funcionamento de cada método, pontuando vantagens e desvantagens na utilização dos mesmos e destacando alguns trabalhos que se utilizam dessa abordagem para estimar a localização de nodos em RSSFs.

#### 2.2.2.1 Tempo de Chegada – *Time of Arrival (TOA)*

O método TOA (do inglês *Time of Arrival*) (WELLENHOFF, LICHTENEGGER e COLLINS, 1997), utilizado pelo Sistema de Posicionamento Global, baseia-se no tempo ne-

cessário para que um sinal emitido por um nodo transmissor com localização conhecida (âncora) alcance um nodo desconhecido cuja posição deve ser determinada. Esse intervalo de tempo, conhecido como tempo de propagação do sinal, é então multiplicado pela velocidade de propagação do sinal a fim de se obter a distância entre transmissor e receptor. Calculadas as distâncias em relação a, no mínimo, três nodos âncoras, o nodo desconhecido é capaz de determinar a sua localização em um espaço de duas dimensões através do método geométrico de trilateração (multilateração em que existem apenas três pontos de referência) (MULLER, 2009). Na Figura 5, as distâncias calculadas são representadas por  $R_1$ ,  $R_2$  e  $R_3$ , que, neste caso, correspondem ao alcance dos sinais de rádio dos nodos âncoras.

Uma vez que o método em questão baseia-se em informações de tempo para a medição das distâncias, existe a necessidade de sincronização entre os relógios dos nodos que integram a rede, caso contrário os cálculos podem retornar grandes erros de localização. Em muitas aplicações isso pode representar um ponto negativo para a utilização da técnica.

Dentre os algoritmos de localização em RSSFs que se utilizam das medições de TOA, cita-se o esquema proposto em (SATHYAN e HEDLEY, 2013), onde os autores baseiam-se no conceito de localização cooperativa (PATWARI, ASH, *et al.*, 2005) para estimar as trajetórias dos nodos desconhecidos dentro da rede.

#### 2.2.2.2 *Diferença do Tempo de Chegada – Time Difference of Arrival (TDOA)*

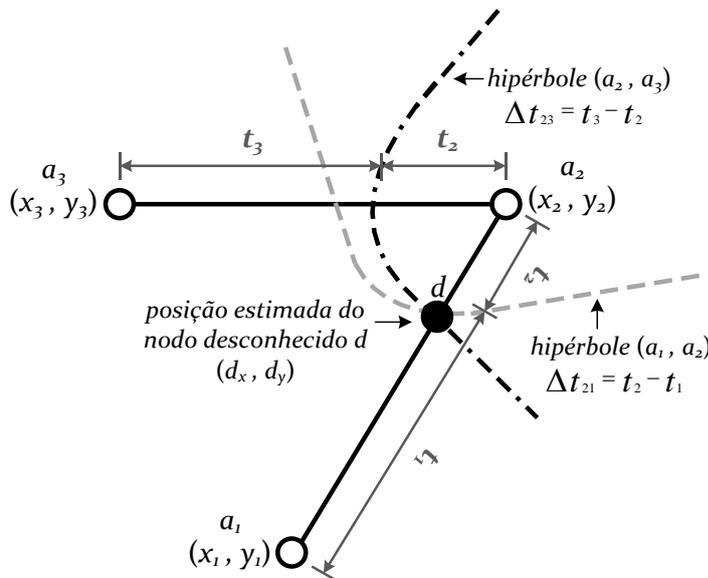
A técnica TDOA (do inglês *Time Difference of Arrival*) (GUSTAFSSON e GUNNARSSON, 2003), também conhecida como técnica hiperbólica de localização, utiliza a diferença entre os tempos necessários para que sinais emitidos por dois nodos âncoras perfeitamente sincronizados alcancem um nodo desconhecido cuja posição deseja-se determinar. A partir da diferença verificada entre os tempos de chegada determina-se uma hipérbole sobre a qual se estima que o nodo desconhecido esteja localizado. Uma hipérbole consiste em uma curva tal que a diferença entre as distâncias de qualquer ponto sobre a curva em relação a dois pontos fixos (focos) é constante. Para o caso específico da técnica TDOA, essa hipérbole caracteriza-se por manter a diferença entre as distâncias dos pontos que a compõem até dois nodos âncoras de referência como sendo constante e obtida a partir da multiplicação da diferença entre os tempos de chegada pela velocidade de propagação do sinal, conforme ilustrado na Figura 6 e representado matematicamente na Equação 3:

$$v \cdot \Delta_{12} = \sqrt{(x_1 - d_x)^2 + (y_1 - d_y)^2} - \sqrt{(x_2 - d_x)^2 + (y_2 - d_y)^2} \quad (3)$$

onde  $v$  é a velocidade de propagação do sinal,  $\Delta_{12}$  é a diferença entre os tempos de chegada dos sinais emitidos pelos nodos âncoras  $a_1$  e  $a_2$  até o nodo desconhecido  $d$ , as coordenadas  $(x_1, y_1)$  e  $(x_2, y_2)$  indicam as posições dos âncoras  $a_1$  e  $a_2$ , respectivamente, e a coordenada  $(d_x, d_y)$  a posição do nodo desconhecido.

Conforme mostra a Figura 6, cada medição de TDOA remonta a uma curva hiperbólica sobre a qual o nodo desconhecido  $d$  pode estar localizado, de tal forma que o ponto de intersecção entre duas dessas curvas corresponde à localização de  $d$ . Na verdade, para um espaço bidimensional, caso nenhuma restrição de área seja imposta à rede, é possível encontrar duas soluções com apenas duas hipérbolas (SPENCER, 2007), e uma terceira curva pode ser necessária para realizar o cálculo de forma mais precisa e definir o ponto único de intersecção entre as curvas (localização do nodo desconhecido). Quando múltiplos nodos âncoras estão disponíveis na região de alcance do rádio de um nodo desconhecido, múltiplas hipérbolas são formadas e a intersecção desse grupo de hipérbolas provê a as coordenadas do nodo desconhecido.

Figura 6 - Estimativa das coordenadas  $(d_x, d_y)$  de um nodo desconhecido  $d$  através do método TDOA.



Fonte: Adaptação de (OLIVEIRA, 2009).

O principal problema associado aos métodos de medição de distâncias baseados no tempo de propagação do sinal (TOA e TDOA) reside no sincronismo necessário entre os relógios dos nodos. Para o caso ilustrado na Figura 6, por exemplo, é necessário que os pares de nodos âncoras  $a_1a_2$  e  $a_2a_3$  estejam sincronizados e emitam seus sinais ao mesmo tempo, caso contrário os erros de estimativa podem se tornar inaceitáveis. Isso pode ser evidenciado para o caso da medição do tempo de propagação de um sinal RF, cuja velocidade de propagação de aproximadamente 300.000km/s faz com que o sinal percorra uma distância de 1m em 3,3ns. Nesse caso, considerando que o relógio do sistema do nodo opere a uma frequência de 10MHz, com o atraso de apenas um ciclo obtém-se um erro de 30m na estimativa de distância através do método TOA. Segundo Savarese (SAVARESE, RABAEY e BEUTEL, 2001), para uma boa estimativa de localização através da técnica TDOA é necessária uma precisão mínima de 0,3ns por metro entre dois nodos âncoras, o que corresponde a um erro de 10%.

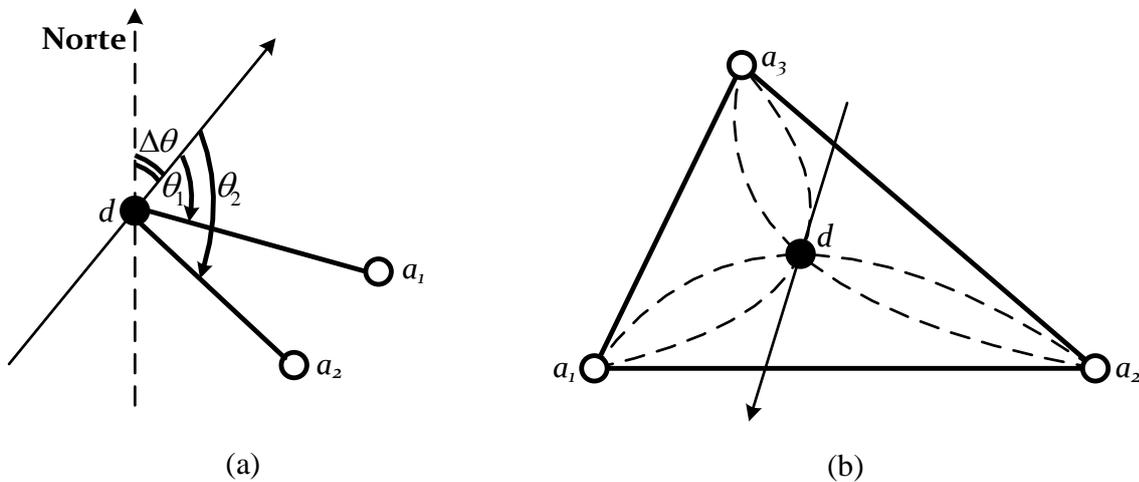
Diante de tais problemas, em (PENG, SHEN e ZHANG, 2012), por exemplo, os autores propõem a utilização de sinais de rádio, que se propagam em velocidades muito menores, porém recaem no problema da necessidade de um hardware especializado. Já em (ZHANG, LI, *et al.*, 2008), apresenta-se uma modificação na técnica TDOA a fim de eliminar a necessidade de sincronização entre os nodos da rede.

### 2.2.2.3 Ângulo de Chegada – *Angle of Arrival (AOA)*

O ângulo de chegada (AOA, do inglês *Angle of Arrival*) é definido como sendo o ângulo formado entre a direção de propagação de uma onda incidente e uma direção de referência, conhecida como orientação (RONG e SICHITIU, 2006). A orientação é, portanto, uma direção fixa em relação à qual o AOA é obtido, sendo medida no sentido horário a partir do Norte. Caso a orientação coincida com a direção Norte ( $0^\circ$  em relação ao Norte), o ângulo de chegada medido é dito absoluto, caso contrário, diz-se que o ângulo é relativo. Para o caso de redes de sensores, uma das formas mais comuns de se obter medições de AOA é através da utilização de um conjunto de antenas direcionais em cada um dos nodos sensores. Outras técnicas são discutidas em (NICULESCU e NATH, 2003) e (NASIPURI e LI, 2002).

Assumindo que os nodos âncoras não possuem informações acerca de suas orientações, a localização dos nodos desconhecidos dentro da rede pode ser obtida a partir de dois métodos que fazem uso da triangulação, conforme ilustra a Figura 7: um deles para o caso em que os nodos desconhecidos conhecem suas orientações, e o outro para o caso em que desconhecem.

Figura 7 - Triangulação no método AOA. (a) Nodo desconhecido  $d$  conhece sua orientação; (b) Nodo desconhecido  $d$  desconhece sua orientação.



Fonte: Adaptação de (RONG e SICHITIU, 2006).

Para o caso ilustrado na Figura 7(a), onde o nodo desconhecido  $d$  conhece sua orientação  $\Delta\theta$ , os ângulos  $\theta_1$  e  $\theta_2$  correspondem aos ângulos de chegada formados entre a orientação do nodo  $d$  e a direção de propagação dos sinais de rádio emitidos pelos nodos âncoras  $a_1$  e  $a_2$ , respectivamente. Uma vez que a orientação de  $d$  é  $\Delta\theta$ , os AOA absolutos medidos a partir de  $a_1$  e  $a_2$  podem ser calculados como  $(\theta_i + \Delta\theta) \pmod{2\pi}$ , com  $i = \{1, 2\}$ . Cada AOA absoluto medido restringe a localização de  $d$  a um segmento de reta iniciando no nodo âncora correspondente à medição. Assim, as coordenadas de  $d$  estão localizadas no ponto de intersecção de todos os segmentos de reta quando  $d$  possui dois ou mais âncoras não colineares em sua vizinhança.

Já para a situação mostrada na Figura 7(b), onde  $d$  desconhece sua orientação e, por isso, os ângulos absolutos não podem ser medidos, utilizam-se as diferenças entre os AOA verificados. Nesse caso, os ângulos  $\angle a_1da_2$ ,  $\angle a_1da_3$  e  $\angle a_2da_3$  podem ser computados a partir dos ângulos de chegada relativos. Cada ângulo  $\angle a_xda_y$  restringe a posição de  $d$  ao arco que passa pelos pontos  $a_x$ ,  $d$  e  $a_y$ . Assim, a localização do nodo desconhecido  $d$  é obtida no ponto de intersecção de todos os arcos quando  $d$  possui três ou mais âncoras não colineares em sua vizinhança.

Em resumo, quando o nodo desconhecido possui informações acerca de sua orientação, a determinação de sua posição através da medição de AOA só é possível se existem pelo menos dois nodos âncoras não colineares em sua vizinhança. Caso o nodo desconhecido não tenha informações de sua orientação, são necessários pelo menos três âncoras não colineares.

Em (NICULESCU e NATH, 2003) e (BISWAS, AGHAJAN e YE, 2005) são apresentados algoritmos de localização de nodos em RSSFs que se utilizam da técnica de medição do AOA.

Como principal desvantagem desta técnica de medição de distâncias tem-se a necessidade de um *hardware* adicional para a determinação do ângulo de chegada do sinal, o qual consiste em um conjunto de antenas direcionais, conforme já mencionado. Além disso, problemas de imprecisão nas medidas ocorrem em função de efeitos como reflexão, difração e dispersão do sinal propagado. Por outro lado, diferentemente das técnicas TOA e TDOA, o método AOA não necessita de sincronização de tempo entre os nodos.

#### 2.2.2.4 Indicação de Intensidade do Sinal Recebido – Received Signal Strength Indicator (RSSI)

O método de medição de distâncias conhecido como RSSI (do inglês *Received Signal Strength Indicator*) baseia-se na potência do sinal recebido (MONDINELLI e KOVACSVAJNA, 2004) (PATWARI e III, 2003). Através de modelos de propagação do sinal de rádio é possível calcular a distância entre o transmissor e o receptor de acordo com a amplitude do sinal recebido, conforme (RAPPAPORT, 2001). A Equação 4, por exemplo, conhecida como fórmula de transmissão de Friis, descreve matematicamente a propagação de ondas eletromagnéticas no espaço livre:

$$\frac{P_r}{P_t} = \left( \frac{\lambda}{4\pi d} \right)^2 \cdot G_t \cdot G_r = L_{fs} \cdot G_t \cdot G_r \quad (4)$$

onde  $P_r$  e  $P_t$  são, respectivamente, as potências recebida e transmitida,  $G_t$  e  $G_r$  são, nessa ordem, os ganhos das antenas do transmissor e do receptor,  $\lambda$  é o comprimento da onda eletromagnética propagada e  $d$  é a distância entre transmissor e receptor.  $L_{fs}$  é a variável que representa a perda de transmissão no espaço livre (do inglês *free space transmission loss*). Assim, estimadas as distâncias entre um nodo desconhecido e, no mínimo, três nodos âncoras, é possível determinar as coordenadas do nodo desconhecido em um espaço bidimensional através do método de trilateração.

Em (DIL, DULMAN e HAVINGA, 2006) e (CHEN, NI, *et al.*, 2011) são apresentados algoritmos de localização/rastreamento de nodos baseados na medição de distâncias através do método RSSI.

Dentre os problemas associados ao método RSSI, destaca-se a imprecisão nas estimativas devido à natureza da propagação do sinal de rádio, o qual sofre inúmeras variações em função de efeitos como reflexão, difração, interferência e falta de visada direta entre transmissor e receptor, principalmente em ambientes internos. Diante disso, com o intuito de minimizar a vulnerabilidade do sinal de RSSI aos fatores mencionados, são propostas alternativas como o pré-processamento dos dados de RSSI coletados (CABRERA-MORA e XIAO, 2008) e modificações no *hardware* do receptor (GRAEFENSTEIN e BOUZOURAA, 2008).

Apesar dos problemas citados, a medição do RSSI é prevista pelo padrão IEEE 802.15.4, conforme (IEEE, 2003), o qual é utilizado em praticamente todos os módulos de comunicação para RSSFs. Assim, o RSSI é o método *range-based* com menor custo de implementação, uma vez que não necessita de *hardware* adicional e, assim como a técnica AOA, não exige sincronização de tempo entre os nodos.

Um exemplo de módulo de comunicação com o padrão IEEE 802.15.4 é o circuito de rádio modelo CC2420 utilizado no nodo comercial MICAz, (TEXAS INSTRUMENTS, 2012) e (CROSSBOW TECHNOLOGY INC., 2013), respectivamente. Esse padrão foi desenvolvido pelo Instituto de Engenheiros Eletricistas e Eletrônicos (IEEE, do inglês *Institute of Electric and Electronic Engineers*) a fim de regulamentar as redes de comunicação sem fio formadas por dispositivos com baixa taxa de transmissão, baixo custo e com alcance limitado, sendo estas as principais características dos nodos.

#### 2.2.2.5 *Discussão acerca dos métodos range-based apresentados*

Conforme pode ser observado na apresentação dos métodos *range-based*, tais técnicas não são capazes, por si só, de determinar as coordenadas dos nodos em uma rede de sensores. Através do emprego de métodos *range-based*, na verdade, é possível apenas estimar distâncias (ângulos, no caso do método AOA) entre nodos da rede. Somente por meio da utilização de técnicas como a trilateração (TOA, TDOA e RSSI) ou a triangulação (AOA) sobre as medições realizadas é que se pode determinar as coordenadas dos nodos. Assim, a escolha entre um ou outro método *range-based* a ser utilizado na implementação de determinado algoritmo de localização não influencia de forma direta na lógica de funcionamento desse algoritmo, uma vez que qualquer uma das técnicas é capaz de realizar as medições necessárias para estimar as localizações dos nodos. A opção por uma das técnicas, então, é normalmente balizada por aspectos como o custo de implementação, a precisão das medições, a necessidade de sincronização entre os relógios do sistema, a necessidade de um hardware específico e/ou a

quantidade de nodos âncoras necessários para executar o método geométrico de posicionamento.

Como já mencionado, o principal problema associado aos métodos *range-based* baseados no tempo de propagação do sinal de rádio, TOA e TDOA, reside no sincronismo necessário entre os relógios dos nodos que integram a rede. Nesse caso, o uso de relógios imprecisos pode conduzir a erros de medição inaceitáveis. Diante disso, algumas alternativas são propostas com o objetivo de reduzir ou até mesmo eliminar a dependência de sincronização entre relógios, como as apresentadas em (KANG, NAMGOONG, *et al.*, 2006) e (ZHANG, LI, *et al.*, 2008). Além disso, os métodos baseados na medição de tempo necessitam de, pelo menos, três nodos âncoras para localizar um nodo desconhecido através do método de multilateração.

Já o método que se utiliza da medição do ângulo de chegada do sinal, AOA, é o mais caro dentre os métodos apresentados, uma vez que necessita de um hardware específico (conjunto de antenas direcionais) integrado a cada um dos nodos da rede. Por outro lado, como já mencionado, quando os nodos desconhecidos que integram a rede possuem conhecimento acerca de suas orientações, o método AOA necessita de apenas dois nodos âncoras para executar a triangulação e determinar a localização dos nodos.

Por fim, dentre as técnicas expostas, a técnica RSSI é a de menor custo econômico, uma vez que não necessita de hardware específico. A precisão das medições obtidas através desse método, no entanto, é baixa quando comparada à precisão dos demais devido à natureza de propagação do sinal de rádio, o qual sofre variações em função de efeitos como reflexão, difração, interferência e falta de visada direta entre transmissor e receptor, principalmente em ambientes internos. Nesse sentido, objetivando minimizar a vulnerabilidade do sinal de RSSI aos fatores mencionados, são propostas alternativas como as apresentadas em (CABRERA-MORA e XIAO, 2008) e (GRAEFENSTEIN e BOUZOURAA, 2008).

### 2.2.3 Métodos *range-free*

Em algoritmos *range-free*, as coordenadas dos nodos desconhecidos são estimadas a partir do conteúdo de mensagens trocadas entre esses nodos e os nodos âncoras que integram a rede. Embora a grande quantidade de mensagens trocadas por esses algoritmos represente um aumento no consumo de energia, há casos em que essas mensagens podem ser aproveitadas por outras aplicações que necessitam trocar informações para o seu funcionamento (MARTINS, NUNES, *et al.*, 2013).

Apesar de o uso de técnicas *range-based* contribuir para um aumento na precisão das estimativas de localização dos nodos, os algoritmos que se utilizam de uma abordagem *range-free* apresentam, muitas vezes, um melhor custo-benefício para a aplicação em questão, principalmente em função do custo adicional exigido pelo hardware específico responsável pelas medições de distância na maioria dos algoritmos *range-based* (AMMARI, 2014) (VIVEKANANDAN e WONG, 2006) (HE, HUANG, *et al.*, 2003). Além disso, para algumas aplicações, principalmente em ambientes externos, a precisão menos acurada de algoritmos *range-free* é suficiente, visto que pequenos erros de estimativa, nesses casos, não afetam de maneira significativa o funcionamento da aplicação.

De acordo com (OLIVEIRA, 2009), os algoritmos de localização *range-free* podem ser divididos em:

- técnicas locais, que se utilizam de uma alta densidade de nodos âncoras dentro da área da rede, de tal forma que cada nodo da rede possui uma quantidade significativa de nodos de referência ao alcance de seu rádio, como é o caso dos algoritmos CAB (VIVEKANANDAN e WONG, 2006) e Centroid (BULUSU, HEIDEMANN e ESTRIN, 2000).
- técnicas baseadas na contagem de saltos ou *hop counting* (NICULESCU e NATH, 2001) (HSIEH e WANG, 2006), nas quais as mensagens iniciadas pelos nodos âncoras se propagam dentro da rede com o acréscimo na contagem de saltos ou *hops*. Esse processo é conhecido como *flooding*.

O método de localização de Monte Carlo para RSSFs (HU e EVANS, 2004), por sua vez, consiste em uma abordagem *range-free* operando sobre um cenário de rede bastante dinâmico (nodos âncoras e nodos desconhecidos movem-se livremente) e sem fazer uso da técnica de contagem de saltos para realizar as estimativas de localização. Dessa forma, o algoritmo proposto por Hu e Evans (HU e EVANS, 2004) não é classificado como uma técnica local ou como uma técnica baseada na contagem de saltos. Dada a importância desse método de localização dentro da proposta do presente trabalho, o mesmo é apresentado em detalhes no Capítulo 3.

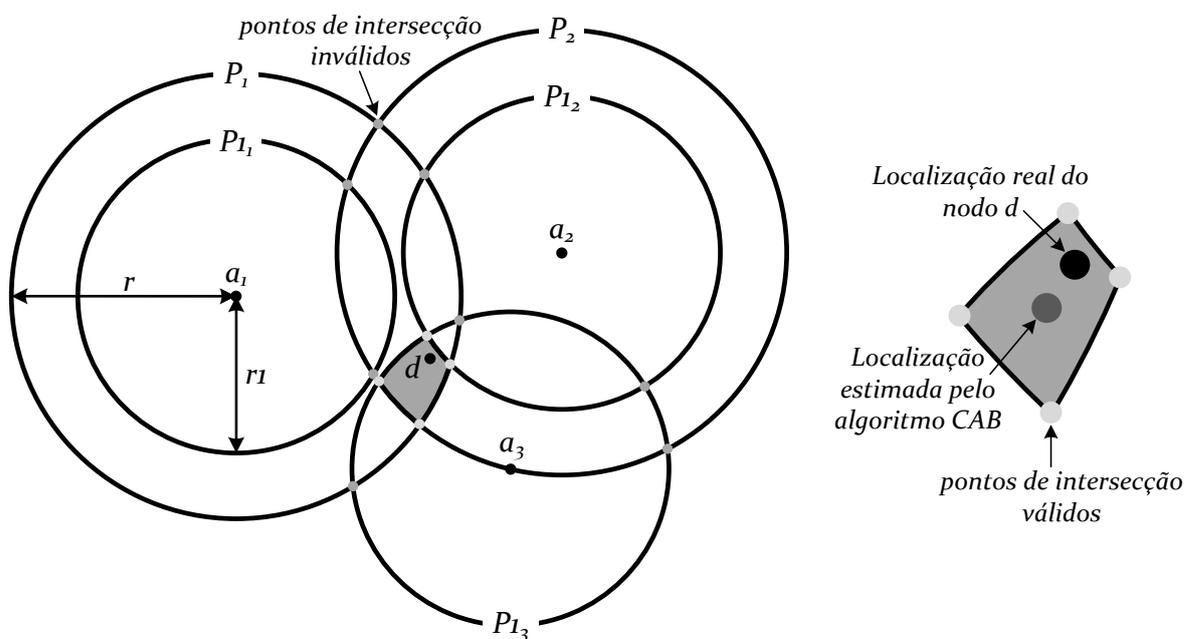
### 2.2.3.1 Técnicas locais

A presente seção introduz algumas técnicas locais de localização *range-free* encontradas no meio científico, apresentando o funcionamento dos algoritmos, pontuando vantagens e

desvantagens na utilização dos mesmos e destacando o aspecto da mobilidade dos nodos dentro da rede.

O algoritmo de localização CAB (VIVEKANANDAN e WONG, 2006) (do inglês *Concentric Anchor-Beacons*) apresenta uma técnica de computação local *range-free* aplicada a redes de sensores estacionárias. A principal diferença do algoritmo CAB em relação a outros algoritmos de localização *range-free* reside no fato de os nodos âncoras transmitirem sinais em diferentes níveis de potência, conforme ilustra a Figura 8. Esses sinais, chamados *beacons*, são transmitidos periodicamente pelos nodos âncoras em dois níveis diferentes de potência.

Figura 8 - Algoritmo CAB: dois níveis de potência são utilizados para determinar os pontos de intersecção válidos e estimar a localização de nodos desconhecidos em uma abordagem *range-free*.



Fonte: Adaptação de (VIVEKANANDAN e WONG, 2006).

Cada *beacon* transmitido pelos nodos âncoras da rede inclui informações acerca da identificação e da localização do transmissor, além do nível de potência associado ao sinal. Assim, com base nos *beacons* recebidos, um nodo desconhecido é capaz de delimitar a região onde está localizado. No caso da Figura 8, por exemplo, o nodo desconhecido  $d$  recebe os *beacons*  $P_1$  e  $P_2$  emitidos, respectivamente, pelos âncoras  $a_1$  e  $a_2$ , e o *beacon*  $P_{1_3}$  emitido pelo âncora  $a_3$  (consequentemente,  $d$  também recebe um *beacon*  $P_3$  não demonstrado na Figura 8). Dessa forma, a cada dois nodos âncoras escutados, o nodo  $d$  define pontos de intersecção ( $x_i$ ,

$y_i$ ) com base nas informações de potência dos *beacons* recebidos, selecionando pontos de intersecção válidos com base na Equação 5.

$$\begin{aligned} r_1 &\leq \sqrt{(x_1 - x_i)^2 + (y_1 - y_i)^2} \leq r \\ r_1 &\leq \sqrt{(x_2 - x_i)^2 + (y_2 - y_i)^2} \leq r \\ \sqrt{(x_3 - x_i)^2 + (y_3 - y_i)^2} &\leq r_1 \end{aligned} \quad (5)$$

As coordenadas  $x_d$  e  $y_d$  estimadas para o nodo  $d$  são obtidas através do cálculo da média das coordenadas dos pontos de intersecção válidos. Essa descrição acerca do funcionamento do algoritmo CAB também é válida para a situação onde o nodo desconhecido recebe *beacons* de apenas dois nodos âncoras. Por outro lado, caso o nodo desconhecido tenha apenas um âncora dentro do alcance de seu rádio, as coordenadas do ponto  $(x_d, y_d)$  são escolhidas aleatoriamente dentro da região formada pelo anel (região delimitada por  $P$  e  $PI$ ) ou pelo círculo (região delimitada por  $PI$ ) onde o nodo desconhecido se encontra.

A precisão do algoritmo CAB está diretamente associada à quantidade de nodos âncoras dispostos dentro da rede. Essa quantidade pode ser reduzida através do aumento do alcance dos rádios dos nodos âncoras. Essa abordagem, entretanto, reduz a precisão do método, uma vez que a área de intersecção se torna maior.

Outro algoritmo que se utiliza de uma técnica local *range-free* para localizar os nodos desconhecidos dentro de uma RSSF é conhecido como Centroid (BULUSU, HEIDEMANN e ESTRIN, 2000). Nesse algoritmo, os nodos âncoras são supostos serem estáticos e distribuídos de maneira uniforme dentro do cenário, de tal forma que, quando considerados conjuntamente, os alcances de seus sinais de rádio cobrem toda a área da rede. Os nodos desconhecidos, por sua vez, movem-se aleatoriamente.

Durante a execução do algoritmo, cada um dos nodos âncoras realiza transmissões periódicas de mensagens que são recebidas por todos aqueles nodos que se encontram dentro do raio de alcance do âncora transmissor, sendo o período de transmissão adotado conhecido por todos dentro da rede. Esse tipo de transmissão é conhecido como *broadcast* e, para o algoritmo Centroid, o único conteúdo das mensagens é a localização do nodo âncora que está transmitindo e a identificação desse nodo. Dessa forma, dentro de um intervalo de tempo  $t$ , os nodos desconhecidos irão receber as mensagens transmitidas pelos âncoras e contabilizá-las, estabelecendo uma métrica de conexão ( $CM_i$ ) que resulta da divisão do número total de men-

sagens recebidas de um determinado âncora  $i$  ( $N_{recv}(i,t)$ ) pelo número total de mensagens transmitidas por esse âncora ( $N_{sent}(i,t)$ ) dentro do intervalo de tempo  $t$ , conforme mostra a Equação 6.

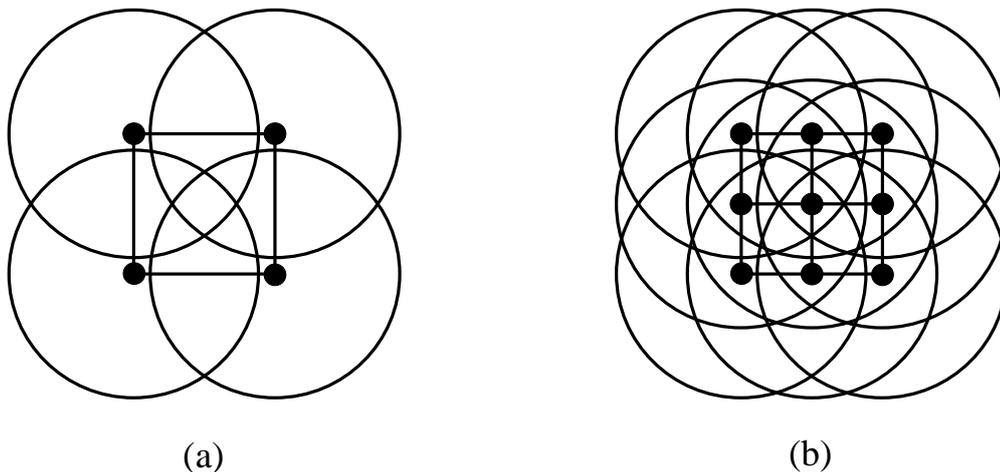
$$CM_i = \frac{N_{recv}(i,t)}{N_{sent}(i,t)} \cdot 100 \quad (6)$$

Por fim, cada nodo desconhecido irá gerar a sua estimativa de localização como sendo a média aritmética das coordenadas dos âncoras cujas métricas de conexão atingiram determinado valor pré-estabelecido ( $CM_{thresh}$ ), conforme mostra a Equação 7:

$$(x_{est}, y_{est}) = \left( \frac{xa_1 + xa_2 + \dots + xa_k}{k}, \frac{ya_1 + ya_2 + \dots + ya_k}{k} \right) \quad (7)$$

onde  $(x_{est}, y_{est})$  é a localização estimada para o nodo desconhecido e  $k$  é o número total de nodos âncoras cujas coordenadas irão participar do cálculo da estimativa. Em (BULUSU, HEIDEMANN e ESTRIN, 2000), o valor de  $CM_{thresh}$  é de 0,9 ou 90%.

Figura 9 - Aumento da densidade de nodos âncoras mantendo o alcance de transmissão. (a) Distribuição regular de 4 nodos âncoras; (b) Distribuição regular de 9 nodos âncoras.



Fonte: Adaptação de (BULUSU, HEIDEMANN e ESTRIN, 2000).

Da mesma forma como ocorre com o algoritmo CAB, segundo (HE, HUANG, *et al.*, 2003), a precisão do algoritmo de localização Centroid é diretamente dependente da densidade de nodos âncoras dispostos dentro da área da rede, conforme ilustram as Figuras 9(a) e 9(b). Isso se deve ao fato de que o aumento dessa densidade acaba aumentando a quantidade

de possíveis localizações estimadas, de forma que a média das posições dos nodos âncoras se torne cada vez mais próxima da localização real do nodo desconhecido, reduzindo o erro de localização (MULLER, 2009). Assim, um dos principais problemas do algoritmo Centroid está associado à topologia da rede, que necessita de uma alta densidade de nodos âncoras distribuídos uniformemente para que se tenha uma precisão razoável nas estimativas de localização.

Por outro lado, dentre as vantagens do modelo Centroid, podemos citar, inicialmente, o fato de que os nodos desconhecidos não necessitam enviar mensagens, o que representa uma economia de energia. Além disso, as mensagens enviadas pelos nodos de referência, conforme já mencionado, são pequenas, contendo apenas a identificação do nodo transmissor e sua localização. Por fim, vale ressaltar que o Centroid não utiliza cálculos complexos, o que reduz consideravelmente o tempo de processamento dos dados e, conseqüentemente, o consumo de energia associado a essa etapa da execução do algoritmo.

### 2.2.3.2 Técnicas de contagem de saltos

A presente seção objetiva elucidar o funcionamento das técnicas de localização *range-free* baseadas na contagem de saltos (do inglês *hops*). Dentre os métodos de contagem de saltos encontrados no meio científico, o algoritmo DV-HOP (NICULESCU e NATH, 2001) é um dos mais conhecidos, servindo de base para a implementação de outros algoritmos (TIAN, ZHANG, *et al.*, 2007) (JI e LIU, 2006) (HSIEH e WANG, 2006) que também se utilizam da contagem de *hops* para estimar a localização de nodos em RSSFs. Assim, tendo em vista a proposta da presente dissertação bem como a importância científica do algoritmo proposto em (NICULESCU e NATH, 2001), esta seção dedica-se à apresentação detalhada do método DV-HOP e de alguns algoritmos baseados nesse método.

O algoritmo DV-HOP é capaz de operar em RSSFs com densidades de nodos âncoras (*landmarks*) muito baixas, de modo que uma grande parcela dos nodos desconhecidos da rede não esteja dentro do raio de alcance de um nodo de referência. A estimativa da localização de um nodo desconhecido através do algoritmo DV-HOP está associada, basicamente, à contagem de saltos que uma mensagem sofre enquanto é propagada dentro da rede. De forma geral, pode-se dividir a execução do algoritmo DV-HOP em três etapas distintas: determinação do fator de correção, contagem de saltos e trilateração (MULLER, 2009).

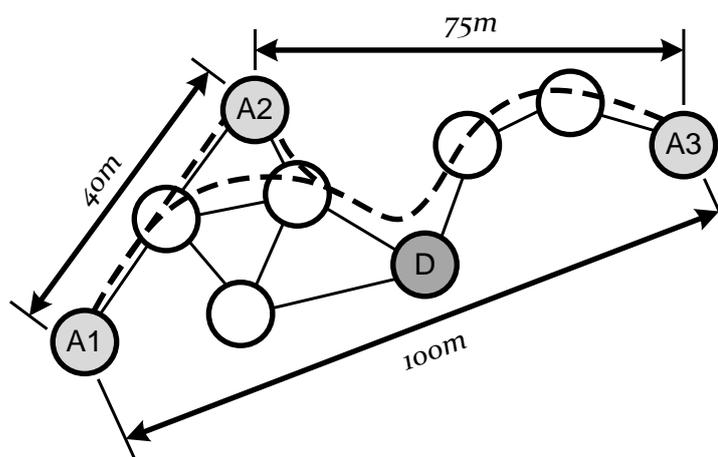
Na primeira etapa, todos os nodos âncoras iniciam a propagação de um contador de saltos através da rede. Cada um dos nodos da rede que recebe a mensagem iniciada por um

âncora incrementa o valor do contador de saltos e retransmite a mensagem àqueles nodos que se encontram dentro do seu raio de alcance, de tal forma que todos os nodos da rede mantêm uma tabela  $\{x_i, y_i, h_i\}$  contendo as coordenadas do nodo âncora  $i$  e a distância em *hops* até esse âncora. Na verdade, os nodos da rede irão receber contadores de saltos com diferentes valores provenientes de um mesmo nodo âncora, devendo, assim, incrementar e retransmitir apenas aqueles contadores que resultam em valores inferiores aos contadores atuais das tabelas armazenadas. Ao término dessa propagação inicial, ou seja, quando os contadores de saltos iniciados pelos âncoras atingem todos os nodos possíveis dentro da rede, cada um dos âncoras, em posse da localização e do número de saltos referentes a cada um dos outros nodos de referência, estima um fator de correção que representa, na verdade, o tamanho médio de um *hop*. A estimativa do fator de correção por um nodo âncora  $i$  é feita através da Equação 8:

$$c_i = \frac{\sum \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum h_i}, i \neq j \quad (8)$$

onde  $j$  refere-se aos âncoras cujos contadores de saltos foram recebidos por  $i$ ,  $x_i$  e  $y_i$  são as coordenadas do nodo âncora  $i$ ,  $x_j$  e  $y_j$  são as coordenadas do nodo âncora  $j$  e  $h_i$  é a distância, em *hops*, entre o âncora  $i$  e o âncora  $j$ .

Figura 10 - Localização do nodo desconhecido  $D$  através do método DV-HOP.



Fonte: Adaptação de (NICULESCU e NATH, 2001).

Através do diagrama ilustrado na Figura 10, é possível exemplificar a estimativa dos fatores de correção por parte dos nodos âncoras  $A1$ ,  $A2$  e  $A3$ . Nesse caso, terminada a propa-

gação inicial dos contadores de saltos, o âncora  $A1$  possui tanto as distâncias em metros, 40m e 100m, quanto as distâncias em *hops*, 2 e 6 *hops*, em relação aos âncoras  $A2$  e  $A3$ , respectivamente. Assim,  $A1$  irá calcular o fator de correção  $c_1$  como sendo  $\frac{40+100}{2+6} = 17.5$ , valor que, na verdade, representa o tamanho médio de um *hop* em metros. Da mesma forma,  $A2$  estima o fator de correção  $c_2 = \frac{40+75}{2+5} = 16.42$  e  $A3$  estima  $c_3 = \frac{75+100}{5+6} = 15.90$ .

Calculados os fatores de correção por cada um dos âncoras da rede, tem início a segunda etapa da execução do algoritmo DV-HOP. Nessa etapa, todos os âncoras, novamente, iniciam a propagação de um contador de saltos através da rede. Dessa vez, no entanto, o fator de correção estimado por cada um dos âncoras é transmitido junto com as respectivas localizações e os respectivos contadores de saltos. Aqui, tomando novamente a Figura 10 como exemplo, os âncoras  $A1$ ,  $A2$  e  $A3$  irão transmitir, respectivamente, os fatores de correção  $c_1$ ,  $c_2$  e  $c_3$ .

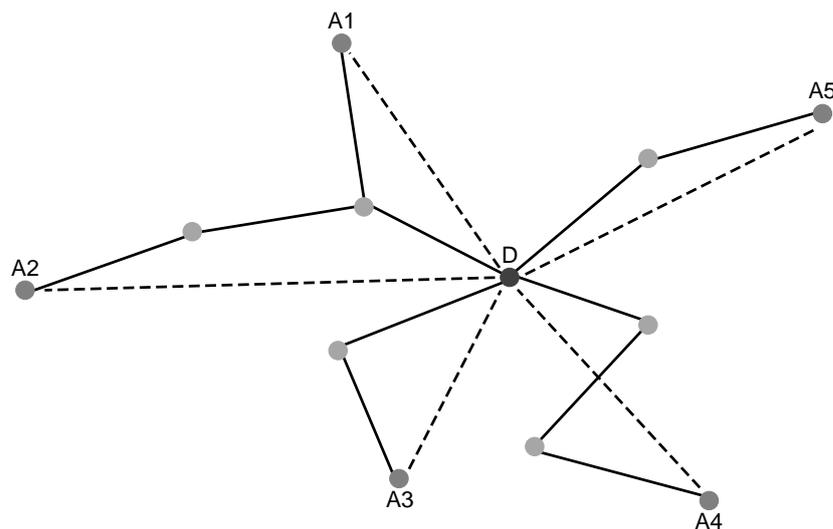
Por fim, na terceira e última etapa, cada um dos nodos desconhecidos da rede gera estimativas de distâncias em relação aos âncoras através da multiplicação dos contadores de saltos pelo fator de correção referente ao âncora mais próximo. Assim, estimadas as distâncias em relação a, pelo menos, três âncoras, os nodos desconhecidos estimam suas localizações através do método de multilateração. No exemplo da Figura 10, para realizar as estimativas de distâncias em relação aos âncoras, o nodo desconhecido  $D$  irá utilizar o fator de correção  $c_2 = 16.42$ , uma vez que  $A2$  é o âncora mais próximo (2 *hops* de distância). Dessa forma, as distâncias estimadas pelo nodo  $D$  em relação aos três nodos âncoras são: em relação a  $A1$ ,  $3 \times 16.42 = 49.26\text{m}$ ; em relação a  $A2$ ,  $2 \times 16.42 = 32.84\text{m}$ ; e em relação a  $A3$ ,  $3 \times 16.42 = 49.26\text{m}$ .

Uma vez que o método DV-HOP aplica-se a RSSFs com disposição aleatória dos nodos dentro do cenário da rede, as estimativas de distâncias a partir da contagem de saltos podem apresentar erros grosseiros (MULLER, 2009), conforme ilustra a Figura 11. Por isso, alguns trabalhos sugerem alterações no algoritmo DV-HOP a fim de obter melhorias na precisão das estimativas através do método.

Em (TIAN, ZHANG, *et al.*, 2007), os autores partem do pressuposto que as estimativas de distâncias feitas pelos nodos desconhecidos em relação a cada um dos âncoras possuem erros em diferentes proporções, de tal forma que o erro ao estimar a localização de um nodo aumenta na medida em que aumenta a quantidade de distâncias estimadas com maiores erros. Na Figura 11, por exemplo, o nodo desconhecido  $D$  irá receber os contadores de saltos referentes aos âncoras  $A1$ ,  $A2$ ,  $A3$ ,  $A4$  e  $A5$ , e, através do método DV-HOP, as estimativas de distâncias em relação a esses cinco nodos âncoras serão utilizadas na multilateração. No en-

tanto, pela análise da Figura 11, observa-se que os caminhos entre o nodo desconhecido  $D$  e os âncoras  $A1$ ,  $A2$  e  $A5$  aproximam-se de uma linha reta, ao passo que os caminhos entre o nodo  $D$  e os âncoras  $A3$  e  $A4$  são bastante sinuosos. Assim, quando se estima as distâncias entre o nodo  $D$  e os nodos  $A3$  e  $A4$  como sendo a distância média em *hops*, e essas estimativas entram no processo de localização do nodo desconhecido, é possível que o resultado apresente um erro significativo. Por isso, em (TIAN, ZHANG, *et al.*, 2007), os autores propõem o algoritmo SANLA (do inglês *Selective Anchor Node Localization Algorithm*), o qual introduz uma etapa de seleção de nodos âncoras ao método DV-HOP, de tal forma que os nodos desconhecidos são capazes de selecionar, dentre todos os âncoras disponíveis, aqueles cujas estimativas de distâncias possuem maior precisão.

Figura 11 - Fonte de erro no algoritmo DV-HOP.



Fonte: Adaptação de (TIAN, ZHANG, *et al.*, 2007).

Outra alternativa é proposta em (SAVARESE, RABAEY e LANGENDOEN, 2002), onde se sugere que o algoritmo DV-HOP seja utilizado para gerar estimativas iniciais das localizações dos nodos desconhecidos e, em seguida, a fim de aumentar a precisão dessas estimativas, inicia-se uma fase de refinamento, na qual ocorre a troca de mensagens contendo as estimativas entre nodos vizinhos.

As simulações e os resultados apresentados para o algoritmo DV-HOP em (NICULESCU e NATH, 2001) consideram uma rede de sensores estacionária. Os autores afirmam, entretanto, que o algoritmo é capaz de operar em uma rede móvel cuja topologia não se altere constantemente.

Nesse sentido, em (CHAO, HAN, *et al.*, 2013), são propostos três modelos de localização a fim de avaliar a performance do algoritmo DV-HOP em RSSFs com nodos âncoras móveis e nodos desconhecidos estacionários. Tais modelos de localização consideram o algoritmo DV-HOP operando em uma RSSF cujos nodos âncoras movimentam-se de acordo com um modelo de mobilidade específico: *Random Waypoint* (RWP) (JOHNSON e MALTZ, 1996), *Random Direction* (RD) (ROYER, MELLIAR-SMITH e MOSER, 2001) ou *Reference Point Group Mobility* (RPGM) (HONG, GERLA, *et al.*, 1999). Os três modelos de mobilidade mencionados são classificados como modelos de mobilidade aleatória, uma vez que as trajetórias percorridas pelos nodos âncoras durante o tempo de simulação são geradas aleatoriamente, considerando as especificidades de cada modelo. O algoritmo DV-HOP, por sua vez, é implementado dentro dos intervalos de pausa<sup>4</sup> na movimentação dos nodos âncoras, ou seja, durante um período de tempo onde a topologia da rede permanece estacionária.

Já em (HSIEH e WANG, 2006), os autores apresentam um método de localização que objetiva adaptar de maneira eficaz o algoritmo DV-HOP a uma RSSF com nodos âncoras e nodos desconhecidos móveis. Até então, técnicas de localização *range-free* baseadas na contagem de saltos não se mostravam eficientes quando aplicadas a RSSFs móveis. Isso acontece, principalmente, porque (NICULESCU e NATH, 2003) as mensagens propagadas pelos nodos âncoras através de longas distâncias podem se tornar desatualizadas ou acumular erros, uma vez que a topologia da rede está sujeita a alterações durante a propagação do sinal. Assim, ao utilizar tais informações para estimar as localizações dos nodos desconhecidos, o algoritmo DV-HOP pode gerar erros inaceitáveis.

Diante disso, Hsieh e Wang (HSIEH e WANG, 2006) propõem um novo algoritmo de localização para RSSFs, chamado DRL (do inglês *Dynamic Reference Localization*), que busca otimizar o funcionamento do método DV-HOP fazendo-o operar localmente. Assim, ao invés de os nodos âncoras propagarem suas mensagens através de toda a área da rede, como acontece no algoritmo DV-HOP, o método DRL determina de maneira dinâmica e individualizada a região de cobertura de cada âncora, limitando a distância, em hops, que as mensagens devem percorrer. Basicamente, o algoritmo DRL possui duas etapas:

- atualização das informações dos nodos âncoras;
- triangulação robusta.

Durante o funcionamento do algoritmo, cada um dos nodos âncoras da rede possui, associado a si, dois parâmetros que são alterados em certa frequência e de maneira dinâmica: o

---

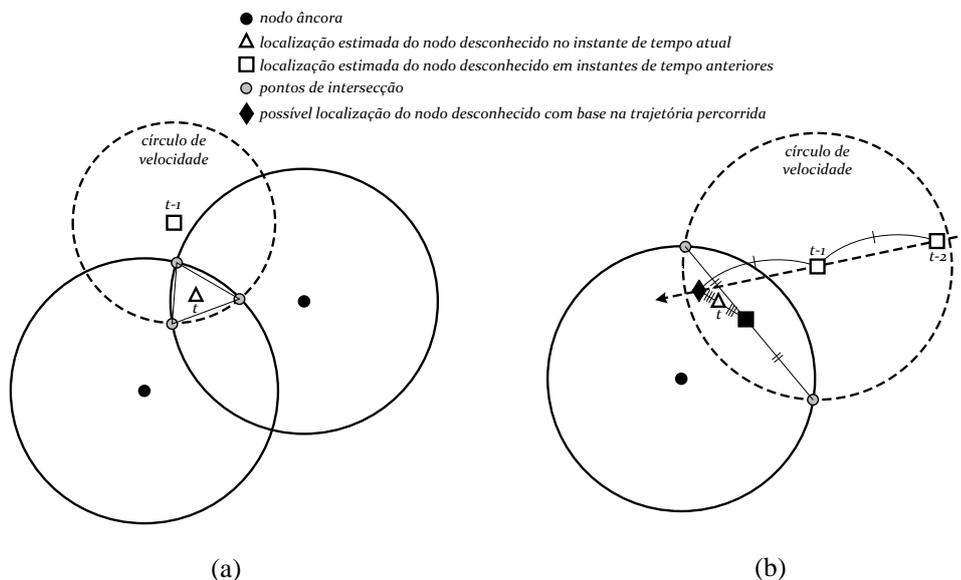
<sup>4</sup> O intervalo (tempo) de pausa consiste no período de tempo que o nodo permanece estacionário entre as mudanças de velocidade e/ou orientação do seu movimento.

fator de correção ( $c$ ) e o *flooding-hop* ( $fh$ ). O fator de correção, conforme já mencionado durante a explanação do método DV-HOP, representa o tamanho médio de um *hop* e é calculado através da Equação 8. Para o caso específico do algoritmo DRL, as variáveis  $x_j$  e  $y_j$  da Equação 8 representam as coordenadas daqueles nodos âncoras cujas mensagens alcançam o nodo âncora  $i$ , dado que o algoritmo opera localmente. O parâmetro *flooding-hop*, por sua vez, indica a distância, em *hops*, que as mensagens propagadas pelos nodos âncoras devem percorrer. O cálculo do parâmetro  $fh$  é feito de acordo com a Equação 9:

$$fh_i = \frac{\sum \frac{h(i,j)}{2}}{n_a}, i \neq j \quad (9)$$

onde  $h(i, j)$  representa a distância, em *hops*, entre o nodo âncora  $i$  e cada um dos âncoras  $j$  cujas mensagens alcançam  $i$ , e  $n_a$  representa o número de nodos âncoras cujas mensagens alcançam  $i$ . Dessa forma, a etapa de atualização das informações dos nodos âncoras corresponde à etapa onde os âncoras da rede atualizam seus respectivos parâmetros  $c$  e  $fh$ .

Figura 12 - Estimativa de localização de um nodo desconhecido através do algoritmo DRL: etapa de triangulação robusta. (a) Situação onde há apenas dois nodos de referência; (b) Situação onde há um único nodo de referência.

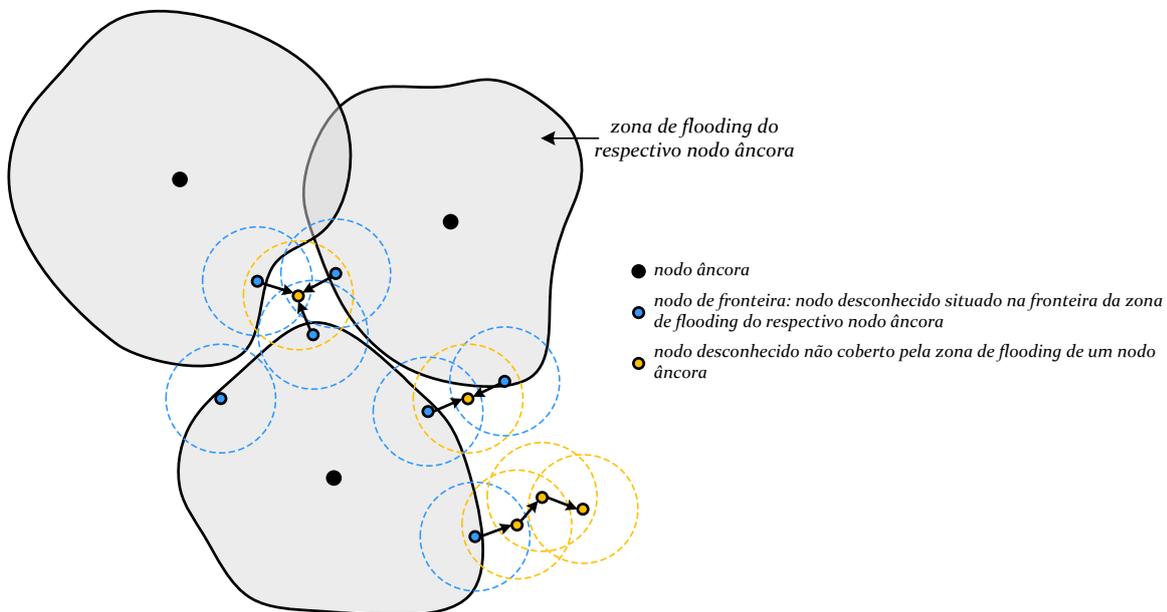


Fonte: Adaptação de (HSIEH e WANG, 2006).

Durante a etapa de triangulação robusta, cada um dos nodos desconhecidos da rede estima sua localização através do método convencional de trilateração (multilateração) ou do

uso de outras informações de referência. Dada a operação local do algoritmo DRL, há situações em que alguns nodos desconhecidos da rede não recebem informações (coordenadas e fator de correção) provenientes de três ou mais nodos âncoras e, dessa forma, não podem estimar seu posicionamento através do método convencional de trilateração, conforme descrito na Seção 2.2.1.1. Diante disso, Hsieh e Wang apresentam algumas soluções para o problema de localização de nodos quando não há a quantidade suficiente de nodos de referência para a realização da trilateração convencional.

Figura 13 - *Backup* inteligente: estimativa de localização de um nodo desconhecido através do algoritmo DRL quando não há nodos de referência.



Fonte: Adaptação de (HSIEH e WANG, 2006).

Para o caso onde há apenas dois nodos de referência, os autores sugerem realizar a trilateração utilizando as duas circunferências centradas nas coordenadas dos nodos âncoras e uma terceira circunferência de raio igual à velocidade máxima que o nodo desconhecido pode assumir (distância máxima que o nodo pode percorrer em uma unidade de tempo) e centro no ponto correspondente à última estimativa de localização desse nodo, conforme ilustra a Figura 12(a). Já para o caso onde há um único nodo de referência, propõe-se utilizar: o ponto médio do segmento de reta que une os dois pontos de intersecção entre as circunferências formadas com base nas informações de localização do nodo âncora e de velocidade máxima do nodo desconhecido; o ponto gerado com base nas duas últimas estimativas de localização do nodo

desconhecido, assumindo que o mesmo move-se em linha reta e com a mesma velocidade. Tal situação é representada na Figura 12(b).

Por fim, na situação onde não há nodos de referência, a estimativa de localização do nodo desconhecido é feita através do processo chamado de *backup* inteligente, conforme ilustra a Figura 13. Nesse caso, o nodo desconhecido em questão solicita informações de localização de nodos âncoras aos nodos desconhecidos que estão em sua vizinhança. Caso esses nodos vizinhos não possuam tais informações, os mesmos irão solicitá-las às suas respectivas vizinhanças, até obterem-nas.

Comparada ao método DV-HOP tradicional, a abordagem local proposta pelo algoritmo DRL reduz o fluxo de mensagens e o consumo de energia da rede, bem como adapta de maneira eficiente a técnica de contagem de saltos a RSSFs móveis.

### 2.2.3.3 *Discussão acerca dos métodos range-free apresentados*

Diferentemente dos algoritmos de localização que utilizam hardware especializado de medição de distâncias (métodos *range-based*) para determinar localizações, os algoritmos *range-free* baseiam-se apenas na conectividade entre os nodos para estimar as coordenadas dos mesmos dentro da rede, empregando diferentes protocolos para isso. Dessa forma, quando comparados a métodos *range-based*, algoritmos *range-free* tendem a ser mais dependentes dos cenários sobre os quais foram projetados.

Diferentes algoritmos são projetados para cobrir distintas situações (LANGENDOEN e REIJERS, 2003) de forma que não existe um método ou algoritmo que seja apropriado para todas as aplicações de redes (OLIVEIRA, 2009), e, nesse sentido, a aplicação e a estrutura topológica da rede (densidade, disposição e mobilidade dos nodos) são fatores extremamente determinantes no processo de escolha entre algoritmos *range-free*. Dessa forma, torna-se difícil a realização de uma análise comparativa entre tais algoritmos sem levar em consideração os fatores mencionados.

Entretanto, permitindo-se realizar uma análise descolada de tais determinantes, um fator importante a ser destacado é a densidade de nodos âncoras necessária para garantir uma precisão satisfatória nas estimativas do algoritmo, uma vez que essa métrica está diretamente relacionada ao custo econômico de implementação da rede. Nesse aspecto, os algoritmos *range-free* baseados em técnicas de contagem de saltos levam vantagem sobre os algoritmos que utilizam técnicas locais, tendo em vista que os primeiros, por valerem-se da disseminação de

informação entre os nodos da rede para estimar suas localizações, são capazes de operar em redes com densidades de nodos âncoras muito baixas.

Considerando o aspecto da mobilidade da rede, destaca-se a versatilidade do método *range-free* proposto em (HSIEH e WANG, 2006), dado que o mesmo opera de maneira satisfatória tanto em RSSFs estacionárias quanto em RSSFs móveis. Além disso, o fato de implementar o método DV-HOP localmente faz com que o algoritmo DRL reduza o fluxo de mensagens e, conseqüentemente, o consumo de energia da rede quando comparado aos métodos tradicionais baseados na contagem de saltos.

#### **2.2.4 O impacto da mobilidade sobre algoritmos de localização em RSSFs**

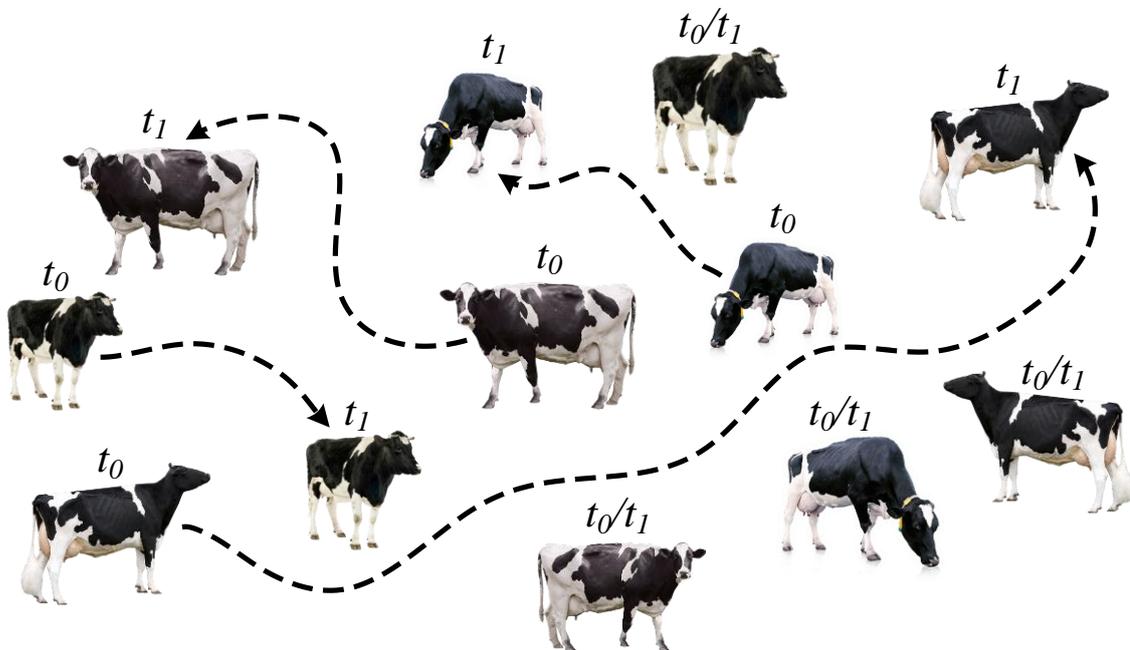
Conforme já mencionado, as aplicações envolvendo RSSFs podem incluir cenários onde a topologia da rede permanece inalterada ao longo do tempo (redes estacionárias), bem como cenários onde a rede apresenta alterações topológicas em determinada frequência (redes móveis). Em outras palavras, tais cenários podem considerar nodos âncoras e/ou desconhecidos como sendo dispositivos fixos ou móveis dentro da rede.

O aspecto da mobilidade, por sua vez, influencia de maneira significativa a implementação de algoritmos de localização e de rastreamento em RSSFs, uma vez que a alteração topológica da rede exige que as localizações dos nodos sejam atualizadas em uma frequência próxima àquela em que a estrutura da rede se altera, introduzindo uma componente em tempo real ao algoritmo, conforme ilustra a Figura 14. Dentre os algoritmos *range-free* apresentados na Seção 2.2.3, por exemplo, aparece o CAB, que se aplica a redes com nodos âncoras e nodos desconhecidos fixos, o Centroid, que considera nodos âncoras fixos e nodos desconhecidos móveis, e o DV-HOP, que, segundo os autores, aplica-se tanto a redes estacionárias como a redes móveis, apresentando variação na precisão das estimativas em função do cenário considerado. Nos três métodos citados, a lógica de funcionamento dos algoritmos e/ou a precisão das estimativas dos mesmos é diretamente influenciada pela mobilidade (ou ausência dela) da rede.

A fim de ilustrar tal influência, os algoritmos de localização *range-free* baseados na contagem de saltos são tomados como exemplos. Conforme visto anteriormente, esses métodos utilizam-se da disseminação de informação entre os nodos da rede para estimar as localizações dos mesmos. Assim, considerando uma rede móvel de sensores e o método DV-HOP, no momento em que uma mensagem propagada através da rede atinge determinado nodo que é importante no processo de estimar o tamanho médio de um *hop*, é provável que a topologia

da rede já tenha se alterado. Dessa forma, os nodos desconhecidos, considerando uma abordagem de computação distribuída, irão estimar localizações imprecisas. Segundo (OLIVEIRA, 2009), isso não ocorre em função da falta de informações ou de imprecisões intrínsecas da computação empregada no método, mas sim pela forma como o algoritmo adquire as informações utilizadas nos cálculos. Isso representa, portanto, uma perda da qualidade da informação utilizada pelo método de localização na medida em que esta se propaga através da rede.

Figura 14 - Exemplo do movimento aleatório dos nodos representados pelos animais de um rebanho bovino. Aqui,  $t_0$  e  $t_1$  ( $t_1 > t_0$ ) indicam instantes de tempo distintos, evidenciando a movimentação de alguns nodos da rede e o estado estacionário de outros.



Fonte: Adaptação de (OLIVEIRA, 2009).

Tratando-se de redes móveis de sensores, um fator importante a ser considerado é a forma através da qual os nodos móveis adquirem movimento. Uma abordagem bastante trabalhada no meio científico considera RSSFs cujos nodos âncoras possuem, por meio de atuadores, controle acerca de seus movimentos (AHMED e AVARITSIOTIS, 2014) (YUAN, CHEN e XI, 2006). Nesse caso, os algoritmos de localização e/ou rastreamento procuram determinar a posição de um robô (nodo com atuador de movimento) em uma área com um mapa predefinido (OLIVEIRA, 2009). Esse robô, normalmente, atua de maneira colaborativa com outros nodos fixos e/ou móveis a fim de contribuir no processo de estimar as localizações de nodos desconhecidos e/ou rastrear alvos móveis dentro da área da rede, como em (WANG, PENG,

*et al.*, 2014) e (AHMED e AVARITSIOTIS, 2014). Por outro lado, a estimativa das coordenadas de nodos desconhecidos em RSSFs móveis sem atuadores para o movimento opera em um terreno sem mapeamento prévio e sem qualquer controle sobre a movimentação dos nodos, como é o caso ilustrado na Figura 14. Considerando o aspecto da mobilidade, o último cenário descrito representa o caso mais crítico para a localização de nodos em RSSFs (OLIVEIRA, 2009).

A presença de nodos móveis com atuadores para o movimento agrega vantagens ao processo de localização e/ou rastreamento de nodos dentro de uma RSSF. Nodos com esse tipo de controle são capazes de aumentar a cobertura do fenômeno investigado, movendo-se diretamente para regiões específicas da rede ou mesmo optando por permanecerem na posição em que se encontram (OLIVEIRA, 2009). Além disso, a presença desses nodos contribui para a redução dos chamados buracos dentro da rede. Os buracos constituem regiões onde há uma quantidade insuficiente de nodos tendo em vista uma aplicação específica. No caso de um algoritmo de localização, nodos desconhecidos localizados em buracos dentro da rede não conseguirão estimar suas coordenadas, uma vez que estarão impossibilitados de trocar informações com uma quantidade suficiente de nodos vizinhos. Entretanto, nem sempre a inclusão de robôs a uma RSSF é uma opção viável, principalmente em função dos custos econômicos e do consumo de potência.

Outro aspecto importante a ser levado em consideração quando se trabalha com algoritmos de localização e/ou rastreamento aplicados a redes móveis de sensores diz respeito ao padrão de movimentação dos nodos. A validação de algoritmos de localização sobre redes móveis, seja através de simulações computacionais ou da condução de experimentos práticos, deve prover informações acerca do padrão e da frequência com que cada um dos nodos da rede altera a orientação (direção e sentido) e a velocidade de seus movimentos ao longo do tempo, mesmo quando não há um padrão (modelo de mobilidade aleatória). Em (MARTINS, MARTINS, *et al.*, 2013), por exemplo, os autores apresentam dispositivos que, integrados aos corpos dos bovinos de um rebanho leiteiro, formam uma RSSF cujo padrão de movimentação dos nodos está diretamente associado à rotina diária de ordenha do rebanho. Essa informação acerca do modelo de mobilidade associado aos nodos da rede influencia de maneira significativa no funcionamento de determinado algoritmo de localização, podendo, inclusive, ser incorporada à lógica de funcionamento do algoritmo a fim de auxiliar no processo de localização. Nesse sentido, os autores do algoritmo apresentado em (JUANG, OKI, *et al.*, 2002), realizaram um estudo minucioso acerca do comportamento das zebras (nodos da RSSF considerada) da região de Laikipia, no Quênia, a fim de otimizar o algoritmo de localização proposto

com base nas informações do modelo de mobilidade dos animais. Em relação às considerações feitas no presente parágrafo, em (CAMP, BOLENG e DAVIES, 2002), o autor afirma que a maneira como os dados envolvidos na análise de RSSFs móveis são manipulados para a obtenção dos resultados de desempenho possui uma grande dependência ao uso de modelos de mobilidade como parâmetros de entrada.

### 2.3 RESUMO DO CAPÍTULO

No presente capítulo foram abordados aspectos acerca das características gerais de RSSFs bem como os principais métodos de localização empregados nessas redes. No que diz respeito aos algoritmos de localização apresentados, fez-se a diferenciação entre métodos *range-based* e *range-free*, expondo a teoria relacionada ao funcionamento de cada um dos métodos introduzidos, pontuando as principais vantagens e desvantagens na utilização dos mesmos e destacando o cenário de mobilidade sobre o qual cada técnica foi pensada (redes móveis ou redes estáticas).



### 3 MÉTODO DE MONTE CARLO

O presente capítulo objetiva detalhar os aspectos de funcionamento do método de localização de Monte Carlo, destacando sua aplicação no processo de localização de nodos em RSSFs móveis. Um capítulo exclusivo para tal propósito faz-se necessário pelo fato de o esquema de localização proposto nesta dissertação ser construído sobre o método de localização mencionado.

Assim, nas seções que seguem, são apresentados os fundamentos teóricos necessários à compreensão desse método sequencial de estimativa de localização bem como algumas de suas limitações. Ao final do capítulo, são introduzidos alguns trabalhos científicos desenvolvidos com base no método apresentado, tendo em vista a relevância dos mesmos para o presente estudo.

#### 3.1 INFERÊNCIA BAYESIANA

Conforme já mencionado no presente documento, muitas das aplicações em RSSFs móveis consideram a existência de nodos âncoras que, através da ação de atuadores, possuem controle acerca de seus movimentos. Nesses casos, os algoritmos de localização e/ou rastreamento, em geral, procuram determinar a posição de um ou mais robôs (nodos com atuadores de movimento) dentro de uma área com mapa predefinido. Tais algoritmos de localização, portanto, embora objetivem determinar as posições dos nodos da rede, devem considerar, dentro de sua lógica de funcionamento, métodos que sejam capazes de localizar e rastrear as trajetórias percorridas por robôs.

Apesar de os métodos de localização de nodos em uma rede de sensores assemelhem-se, em diversos aspectos, com as técnicas utilizadas para a localização de robôs, essas duas abordagens apresentam especificidades em função dos cenários sobre os quais são pensadas. Enquanto as aplicações em robótica móvel utilizam-se do mapeamento prévio do cenário considerado, por exemplo, os nodos de uma RSSF movimentam-se em um ambiente desconhecido. Além disso, robôs são normalmente dotados de uma grande quantidade de sensores que os auxiliam no controle de seus movimentos, ao passo que os nodos de uma rede de sensores possuem pouco ou nenhum controle sobre sua mobilidade. Apesar disso, a estrutura de uma RSSF permite que os nodos troquem informações entre si, atuando de forma colaborativa no processo de localização dos dispositivos da rede.

Diante do exposto, esta seção tem por objetivo introduzir os principais conceitos associados ao método probabilístico de localização de Monte Carlo (MCL, do inglês *Monte Carlo Localization*), o qual foi inicialmente definido para o problema de localização e rastreamento de robôs móveis (DELLAERT, FOX, *et al.*, 1999), sendo, algum tempo depois, adaptado para o problema de localização de nodos em RSSFs móveis (HU e EVANS, 2004).

Considerando uma abordagem probabilística, o problema de localização no campo da robótica móvel consiste em estimar o estado de um robô no instante de tempo atual  $k$ , tendo como informações o estado inicial do robô e um conjunto de medições  $Z^k = \{\mathbf{z}_k, i = 1 \dots k\}$  realizadas no instante de tempo  $k$  (DELLAERT, FOX, *et al.*, 1999). Em um sistema de coordenadas de duas dimensões, normalmente trabalha-se com um vetor de estado  $\mathbf{x} = [x, y, \theta]^T$  tridimensional, contendo a posição (coordenadas  $x$  e  $y$ ) e a orientação (ângulo  $\theta$ ) do robô no instante de tempo considerado. Esse problema de estimativa é uma instância da chamada inferência Bayesiana (LADD, BEKRES, *et al.*, 2005) (HARTL e LI, 2005), onde se deseja construir a distribuição posterior<sup>5</sup>  $p(\mathbf{x}_k | Z^k)$  do estado atual  $\mathbf{x}_k$  com base no conjunto  $Z^k$  de medições relacionadas a  $\mathbf{x}_k$ . A ideia de aumento da quantidade de informações sobre  $\mathbf{x}_k$  após a observação de  $Z^k$  é bastante intuitiva e o teorema de Bayes é a regra de atualização utilizada para quantificar esse aumento de informação, conforme descreve a Equação 11. Assim, o problema de localização e rastreamento de um robô móvel, descrito por meio de uma abordagem probabilística, exige a computação recursiva da função densidade de probabilidade (PDF, do inglês *Probability Density Function*)  $p(\mathbf{x}_k | Z^k)$  em cada instante de tempo  $k$ . Esse processo é realizado em duas etapas:

- etapa de predição;
- etapa de atualização ou filtragem.

Durante a etapa de predição, utiliza-se um modelo de mobilidade para estimar a localização atual do robô na forma de uma PDF preditiva<sup>6</sup>  $p(\mathbf{x}_k | Z^{k-1})$ . Aqui, portanto, assume-se que o estado atual  $\mathbf{x}_k$  depende unicamente do estado anterior  $\mathbf{x}_{k-1}$  e de uma entrada de controle  $\mathbf{u}_{k-1}$ , de tal forma que o modelo de mobilidade é descrito como a densidade condicional  $p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_{k-1})$ . Assim, a PDF  $p(\mathbf{x}_k | Z^{k-1})$  é obtida através da Equação 10.

$$p(\mathbf{x}_k | Z^{k-1}) = \int p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_{k-1}) p(\mathbf{x}_{k-1} | Z^{k-1}) d\mathbf{x}_{k-1} \quad (10)$$

<sup>5</sup> Probabilidade condicional obtida através de dados empíricos.

<sup>6</sup> Função que representa probabilisticamente a informação que se tem sobre o estado  $\mathbf{x}_k$  antes da realização das observações em  $Z^k$ .

Já na fase de atualização ou filtragem, utiliza-se um modelo de medição com o intuito de incorporar as informações provenientes de sensores ao processo de obtenção da PDF posterior  $p(\mathbf{x}_k | Z^k)$ . As medições do conjunto  $Z^k$  são condicionalmente independentes do conjunto de medições realizadas no instante de tempo anterior ( $Z^{k-1}$ ), e o modelo de medição considerado é dado em termos da função de verossimilhança<sup>7</sup>  $p(\mathbf{z}_k | \mathbf{x}_k)$  que expressa a probabilidade de o robô estar na localização  $\mathbf{x}_k$  uma vez que  $\mathbf{z}_k$  foi observado. Dessa forma, a PDF  $p(\mathbf{x}_k | Z^k)$  pode ser obtida através do teorema de Bayes, conforme descreve a Equação 11.

$$p(\mathbf{x}_k | Z^k) = \frac{p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | Z^{k-1})}{p(\mathbf{z}_k | Z^{k-1})} \quad (11)$$

Portanto, como uma instância da inferência Bayesiana, o problema de localização e rastreamento na área da robótica móvel é resolvido pela solução recursiva das Equações 10 e 11 para cada instante de tempo  $k$ . Assume-se que o estado inicial  $\mathbf{x}_0$  é disponibilizado na forma de uma densidade  $p(\mathbf{x}_0)$ .

A solução das Equações 10 e 11, por sua vez, pode ser obtida a partir de diferentes métodos existentes, de tal forma que a escolha por um dos métodos depende da forma utilizada para representar a densidade de probabilidade  $p(\mathbf{x}_k | Z^k)$ . Dentre esses métodos, citam-se:

- métodos baseados na aplicação do filtro de Kalman (MAYBECK, 1979);
- métodos baseados no processo estocástico<sup>8</sup> da cadeia de Markov (BURGARD, FOX, *et al.*, 1996);
- métodos baseados em amostragem.

Cada um desses métodos apresenta propriedades bastante específicas que, dependendo da aplicação, podem mostrar-se como vantagens ou empecilhos. Para o estudo do presente trabalho, há um interesse nos métodos baseados em amostragem, nos quais a densidade de probabilidade é representada através de um conjunto de amostras obtidas aleatoriamente de  $p(\mathbf{x}_k | Z^k)$ . Em comparação aos demais métodos citados, métodos baseados em amostragem são de fácil implementação, requerem menores recursos de memória e, em muitos casos, apresentam maior precisão.

---

<sup>7</sup> A diferença entre a função de probabilidade (ou função densidade de probabilidade) e a função de verossimilhança reside na definição de qual termo da função mantém-se fixo e qual varia. Em  $p(\mathbf{x}_k | Z^k)$ , o termo de interesse  $\mathbf{x}_k$  é fixo, enquanto  $Z^k$  varia. Nesse caso, tem-se uma função de probabilidade. Já em  $p(\mathbf{z}_k | \mathbf{x}_k)$ , onde  $\mathbf{z}_k \in Z^k$ , o termo  $\mathbf{z}_k$  é fixo enquanto a variável de interesse  $\mathbf{x}_k$  é variável em relação aos valores paramétricos possíveis. Nesse caso, a função é formalmente chamada de função de verossimilhança.

<sup>8</sup> Grupo de variáveis aleatórias que representam a evolução de determinado sistema ao longo do tempo.

O método de Monte Carlo é um dos algoritmos capazes de resolver o problema da inferência Bayesiana a partir da obtenção de amostras da densidade  $p(\mathbf{x}_k | Z^k)$ . O método foi proposto na década de 70 (HANDSCHIN, 1970) e, com o passar do tempo, foi aplicado em diferentes áreas da ciência (GORDON, SALMOND e SMITH, 1993) (ISARD e BLAKE, 1996) (ISARD e BLAKE, 1998). Uma vez que o algoritmo proposto na presente dissertação tem como base o método de localização de Monte Carlo, a seção que segue destina-se à apresentação desse método.

### 3.2 MÉTODO DE LOCALIZAÇÃO DE MONTE CARLO

Conforme mencionado, os métodos baseados em amostragem representam a densidade de probabilidade através de um conjunto de  $N$  amostras aleatórias ou partículas  $S_k = \{s_k^i; i = 1 \dots N\}$  extraídas da própria densidade  $p(\mathbf{x}_k | Z^k)$ . De acordo com (ISARD e BLAKE, 1998), tal representação é possível em função da dualidade essencial entre as amostras e a densidade a partir da qual elas são geradas.

Assim, quando se trata da utilização de métodos baseados em amostragem para a solução de problemas que recaem na inferência Bayesiana, o objetivo consiste em computar recursivamente, para cada instante de tempo  $k$ , o conjunto de amostras  $S_k$  obtido de  $p(\mathbf{x}_k | Z^k)$ . O método de Monte Carlo, algoritmo que integra a categoria dos chamados filtros de partículas (DOUCET, 1998), é um dos métodos utilizados na solução desse problema descrito. Nesse caso, as etapas de predição e de atualização descritas na seção anterior são realizadas com base nas amostras do conjunto  $S_k$ .

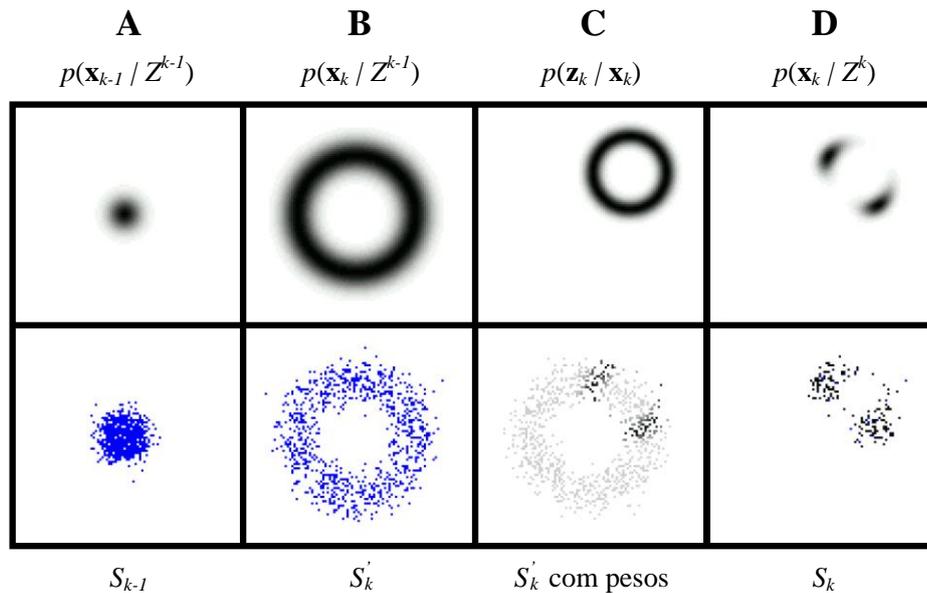
Durante a etapa de predição, de maneira análoga ao procedimento descrito matematicamente na Equação 10, aplica-se um modelo de mobilidade sobre cada uma das amostras  $s_{k-1}^i$  que integram o conjunto  $S_{k-1}$  obtido na iteração anterior (instante de tempo  $k-1$ ). Assim, para cada uma das amostras  $s_{k-1}^i$  obtém-se uma amostra  $s_k^i$  extraída da densidade de probabilidade  $p(\mathbf{x}_k | s_{k-1}^i, \mathbf{u}_{k-1})$ . O conjunto de amostras  $S_k$  gerado, portanto, aproxima, por meio de amostras aleatórias, a PDF preditiva  $p(\mathbf{x}_k | Z^{k-1})$ .

Na fase de atualização ou filtragem, analogamente ao procedimento descrito matematicamente na Equação 11, as medições  $\mathbf{z}_k$  são levadas em consideração, de tal forma que cada uma das amostras em  $S_k$  recebe o peso  $m_k^i = p(\mathbf{z}_k | s_k^i)$ , ou seja, a verossimilhança de  $s_k^i$  dado

que  $\mathbf{z}_k$  foi observado. Assim, o conjunto de amostras  $S_k = \{s_k^j; j = 1 \dots N\}$  que aproxima a PDF posterior  $p(\mathbf{x}_k | Z^k)$  é gerado a partir do conjunto  $\{s_k^i, m_k^i\}$ .

A inicialização do algoritmo, nesse caso, considera um conjunto de amostras  $S_0 = \{s_0^i\}$  que representa a densidade de probabilidade  $p(\mathbf{x}_0)$ .

Figura 15 - Densidades de probabilidade e conjuntos de amostras para uma iteração do método de localização de Monte Carlo aplicado à robótica móvel.



Fonte: (DELLAERT, FOX, *et al.*, 1999).

Para ilustrar as etapas de predição e de filtragem referentes ao método de localização de Monte Carlo, em (DELLAERT, FOX, *et al.*, 1999) os autores apresentam o exemplo gráfico ilustrado na Figura 15. A figura demonstra uma iteração do algoritmo, onde a linha superior traz a densidade de probabilidade exata e a linha inferior traz a representação baseada em amostras (partículas) dessa densidade. No painel A da figura, tem-se o conjunto de amostras  $S_{k-1}$  representando a PDF posterior  $p(\mathbf{x}_{k-1} | Z^{k-1})$  obtida na iteração anterior (instante de tempo  $k-1$ ). No exemplo, conforme pode ser observado, a precisão da estimativa de localização do robô é satisfatória, mas, segundo os autores, sua orientação é desconhecida. Já no painel B, tem-se a densidade de probabilidade  $p(\mathbf{x}_k | Z^{k-1})$  e o respectivo conjunto de amostras  $S'_k$  após a obtenção da informação de que o robô se movera 1 metro desde o último instante de tempo analisado. Nesse caso, sabe-se que o robô se encontra em um círculo de raio igual a 1 metro centrado na sua localização anterior. O painel C ilustra o que acontece quando se observa um ponto de referência localizado a 0.5 metros de distância do robô na direção nordeste. A parte

superior do painel C traz a probabilidade  $p(\mathbf{z}_k | \mathbf{x}_k)$  e a parte inferior mostra como as amostras  $s'_k{}^i$  recebem seus pesos com base nessa probabilidade. Por fim, o painel D ilustra o conjunto  $S_k$  que aproxima a densidade exata  $p(\mathbf{x}_k | Z^k)$  e é obtido a partir do conjunto de amostras já pesadas  $\{s'_k{}^i, m_k^i\}$ . O instante de tempo ilustrado no painel D, por sua vez, corresponde ao início da próxima iteração do algoritmo.

### 3.3 ALGORITMO DE LOCALIZAÇÃO DE MONTE CARLO PARA RSSFs MÓVEIS

Em uma abordagem diferente das apresentadas na Seção 2.2 deste documento, o problema da localização de nodos em uma RSSF móvel pode ser tratado do ponto de vista probabilístico. Nesse caso, da mesma forma como definido para o campo da robótica móvel, deseja-se construir a distribuição posterior  $p(\mathbf{x}_k | Z^k)$  referente ao estado atual  $\mathbf{x}_k$  de cada um dos nodos desconhecidos da rede levando em consideração as observações realizadas no instante de tempo  $k$ . Assim, o problema apresenta-se como uma instância da inferência Bayesiana e, dessa forma, pode ser solucionado por meio do método de Monte Carlo, utilizando-se de amostras para representar as densidades de probabilidade.

Diante disso, em (HU e EVANS, 2004) os autores apresentam um método de localização de nodos em RSSFs móveis baseado no algoritmo descrito na seção anterior (DELLAERT, FOX, *et al.*, 1999). O algoritmo de Monte Carlo para localização de nodos em RSSFs, como é conhecido, consiste em um método *range-free* operando sobre um cenário bastante dinâmico, onde a distribuição dos nodos é irregular, a densidade de nodos âncoras é baixa em comparação a métodos até então existentes, a localização inicial dos âncoras é desconhecida pela rede e tanto nodos âncoras como nodos desconhecidos podem se movimentar de forma aleatória dentro da área da rede. Dessa forma, o método *range-free* em questão não pode ser classificado como uma técnica local ou como uma técnica baseada na contagem de *hops*, conforme categorização apresentada na Seção 2.2.

No algoritmo proposto por Hu e Evans (HU e EVANS, 2004), assume-se o tempo discretizado. Assim, considerando a possibilidade de movimentação dos nodos da rede, existe a necessidade recorrente de estimar as localizações dos nodos a cada unidade de tempo  $k$  (*time step*). Tal problema, portanto, sendo instanciado dentro da inferência Bayesiana, é solucionado através da realização das etapas de predição e de filtragem, como mostra o pseudocódigo do Algoritmo 1.

---

Algoritmo 1 - Pseudocódigo do algoritmo de localização de Monte Carlo para RSSFs.

---

**Inicialização:** Inicialmente, o nodo não tem conhecimento acerca de sua localização.  $N$  é uma constante que indica o número de amostras a serem obtidas.

$$S_0 = \{\text{conjunto formado por } N \text{ amostras aleatórias}\}$$

**Iterações:** Computa um novo conjunto  $S_k$  de possíveis localizações baseado no conjunto  $S_{k-1}$  (conjunto de possíveis localizações obtido para o instante de tempo anterior) e nas novas observações  $Z^k$ .

$$S_k = \{ \quad \}$$

**while** ( $\text{size}(S_k) < N$ ) **do**

**Predição:**  $S'_k = \{s'_k{}^i \mid s'_k{}^i \text{ é selecionada da distribuição de probabilidade } p(\mathbf{x}_k \mid Z^{k-1})$   
(Equação 10) para todo  $1 \leq i \leq N$

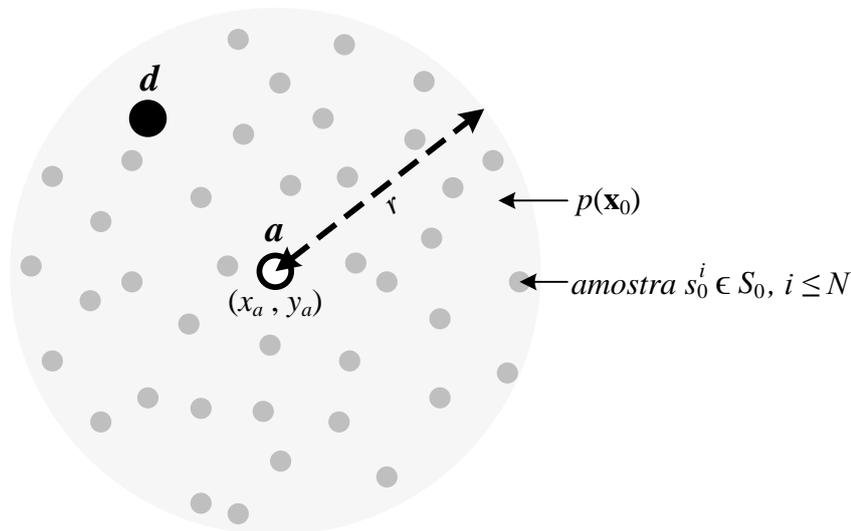
**Filtragem:**  $S_k = \{s_k{}^i \mid s_k{}^i \in S'_k \text{ e } \text{filter}(s_k{}^i) > 0\}$  (Equação 11)

**end**

---

Conforme mencionado na seção anterior, a inicialização do método de localização de Monte Carlo considera um conjunto de amostras  $S_0 = \{s_0^i\}$ ,  $i \leq N$ , que aproxima a densidade de probabilidade  $p(\mathbf{x}_0)$  relacionada ao estado inicial do dispositivo a ser localizado. No caso do algoritmo de localização de Monte Carlo para RSSFs móveis, os nodos desconhecidos não possuem conhecimento acerca de suas coordenadas e orientações, apenas sabem que suas velocidades de movimentação são inferiores ou iguais a um valor máximo definido como  $v_{max}$  e que o alcance de transmissão de seus rádios é  $r$ . Dessa forma, os nodos desconhecidos esperam pela mensagem de algum nodo âncora vizinho informando sua localização. Caso um nodo desconhecido  $d$  descubra a presença de pelo menos um âncora na vizinhança,  $d$  gera o conjunto  $S_0$  com amostras correspondentes a pontos situados no interior da circunferência de raio  $r$  e centro  $(x_a, y_a)$ , onde  $x_a$  e  $y_a$  representam as coordenadas do nodo âncora, conforme ilustra a Figura 16. Em outras palavras, o nodo desconhecido  $d$  posiciona as amostras  $s_0^i$  dentro do raio de alcance do âncora observado.

Figura 16 - Etapa de inicialização do algoritmo de localização de Monte Carlo para RSSFs móveis. Os nodos desconhecidos procuram por nodos âncoras em suas respectivas vizinhanças. Caso verifiquem a presença de pelo menos um âncora, os nodos desconhecidos geram as amostras  $s_0^i$  ( $i \leq N$ ) baseando-se no fato de estarem dentro do raio de alcance do âncora de coordenadas  $(x_a, y_a)$ .



Fonte: Autor.

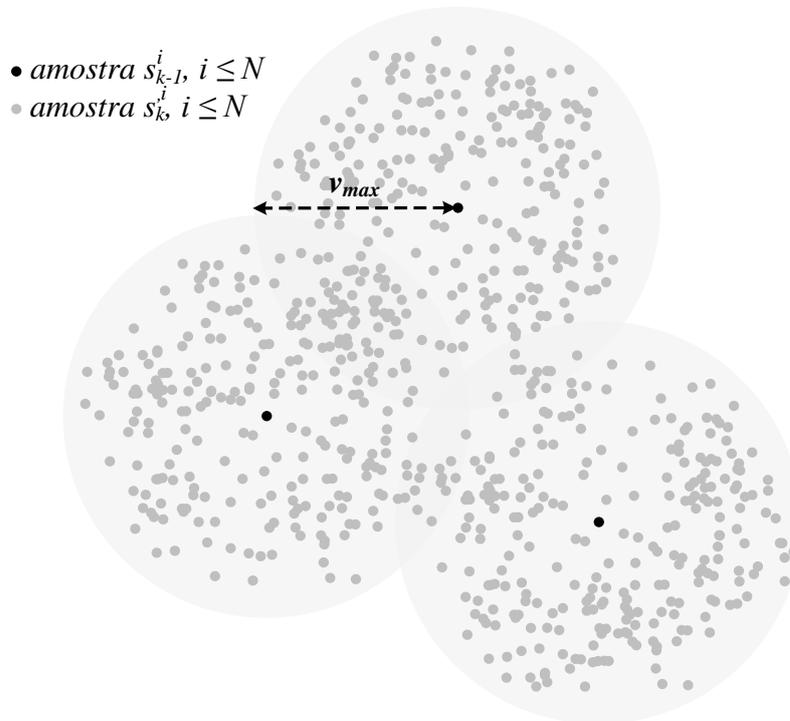
Durante a etapa de predição do algoritmo, da mesma forma como ocorre no processo de localização de um robô (Seção 3.2), aplica-se um modelo de mobilidade sobre cada uma das amostras  $s_{k-1}^i$  que integram o conjunto  $S_{k-1}$  obtido na iteração anterior (instante de tempo  $k-1$ ). Nesse caso, a única informação de que os nodos da rede dispõem é a de que suas velocidades podem variar dentro do intervalo  $[0, v_{max}]$ . Assim, considerando que  $s_{k-1}^i$  é uma possível localização para um dado nodo desconhecido no instante de tempo  $k-1$ , algumas das possíveis localizações  $s_k^i$  para este mesmo nodo no instante de tempo atual  $k$  estão contidas na região circular de raio  $v_{max}$  e centro  $s_{k-1}^i$ , como mostra a Figura 17. Nesse caso, considera-se o tempo discretizado de tal forma que a distância máxima que um nodo pode percorrer entre dois *time steps* consecutivos é  $v_{max}$ , ou seja,  $v_{max}$  representa a distância máxima que um nodo pode percorrer em um intervalo de simulação.

Assumindo que as velocidades distribuem-se de maneira uniforme dentro do intervalo  $[0, v_{max}]$ , a PDF preditiva  $p(\mathbf{x}_k | Z^{k-1})$ , que expressa a probabilidade de o nodo estar na localização  $\mathbf{x}_k$  levando em consideração apenas o estado anterior e o modelo de mobilidade adotado, é dada pela Equação 12 (HU e EVANS, 2004):

$$p(\mathbf{x}_k | Z^{k-1}) = \begin{cases} \frac{1}{\pi v_{max}^2} & \text{se } d(s_k^i, s_{k-1}^i) < v_{max} \\ 0 & \text{se } d(s_k^i, s_{k-1}^i) \geq v_{max} \end{cases} \quad (12)$$

onde  $d(p_1, p_2)$  denota a distância Euclidiana entre dois pontos  $p_1$  e  $p_2$ .

Figura 17 - Etapa de predição do algoritmo de localização de Monte Carlo para RSSFs móveis. Os três pontos representados na cor preta, situados nos centros das circunferências de raios  $v_{max}$ , correspondem a amostras do conjunto  $S_{k-1}$  utilizado para estimar a posição de um nodo qualquer na iteração anterior (instante de tempo  $k-1$ ). Os pontos distribuídos em torno dessas três amostras pertencem ao conjunto  $S'_k$ , formado a partir da aplicação do modelo de mobilidade sobre cada uma das amostras em  $S_{k-1}$ .



Fonte: Autor.

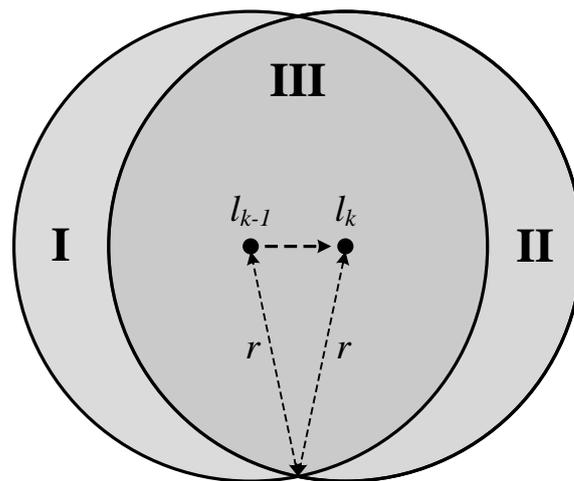
Como pode ser observado, o conjunto  $S'_k$  que aproxima a densidade de probabilidade  $p(\mathbf{x}_k | Z^{k-1})$  é formado por amostras  $s_k^i$  geradas aleatoriamente a partir das informações provenientes do modelo de mobilidade considerado e das amostras  $s_{k-1}^i$ . Isso, portanto, incorpora uma incerteza à localização estimada para o nodo, em função da movimentação aleatória e desconhecida do mesmo.

Diante disso, a etapa de filtragem do algoritmo tem por objetivo reduzir essa incerteza acerca das localizações estimadas, eliminando do conjunto  $S'_k$  aquelas amostras (localizações)

não condizentes com novas informações ( $Z^k$ ) obtidas no instante de tempo atual ( $k$ ). Tais informações estão relacionadas à observação de nodos âncoras, dado que estes possuem conhecimento preciso acerca de suas localizações.

Para fins de simplificação de análise, em (HU e EVANS, 2004) os autores assumem, conforme já mencionado, que o tempo é discreto e que todas as mensagens transmitidas são recebidas instantaneamente dentro da vizinhança. Assim, em um dado instante de tempo  $k$ , todos os nodos situados dentro da zona de alcance de um âncora receberão um anúncio desse nodo de referência, contendo informações sobre sua localização. Obviamente, em uma aplicação real, devem ser considerados os problemas associados à colisão e à perda de mensagens.

Figura 18 - Classificação dos nodos âncoras: o nodo âncora da figura se movimenta da posição  $l_{k-1}$  no instante de tempo  $k-1$  para a posição  $l_k$  no instante de tempo  $k$ . Para nodos situados na região III, o âncora é classificado como *insider*. Para nodos situados na região II, o âncora é classificado como *arriver*. Para os nodos da região I, o âncora é um *leaver*.



Fonte: Adaptação de (HU e EVANS, 2004).

De acordo com essa abordagem, os autores de (HU e EVANS, 2004) consideram quatro tipos de nodos âncoras, conforme ilustra a Figura 18:

- **Outsiders:** âncoras que não se encontram na vizinhança no instante de tempo atual e também não estavam presentes no instante de tempo anterior;
- **Arrivers:** âncoras que se encontram na vizinhança no instante de tempo atual, mas não estavam presentes no instante de tempo anterior;
- **Leavers:** âncoras que se encontravam na vizinhança no instante de tempo anterior, mas não estão presentes no instante de tempo atual;

- **Insiders:** âncoras que se encontravam na vizinhança no instante de tempo anterior e permanecem presentes no instante de tempo atual.

Considerando a categorização apresentada, os âncoras classificados como *arrivers* e *leavers* são aqueles que fornecem as informações mais úteis ao processo de filtragem do algoritmo, uma vez que, nesses casos, um nodo desconhecido  $d$  pode inferir que estava a uma distância máxima  $r$  de  $l_{k-1}$  no instante de tempo  $k-1$ , mas não está a uma distância máxima  $r$  de  $l_k$  no instante de tempo  $k$ , onde  $l$  indica a localização do âncora. Entretanto, ao considerar apenas as informações provenientes dos nodos âncoras, um nodo desconhecido não saberá a localização anterior de um âncora *arriver* ou a localização atual de um *leaver*. Diante desse problema, Hu e Evans apresentam duas alternativas de solução:

1. A cada instante de tempo, um nodo âncora  $a$  transmite a sua localização atual (instante  $k$ ) e a sua localização anterior (instante  $k-1$ ):

$$a \rightarrow \text{Região de cobertura} \quad \text{HELLO} \mid \text{ID}_a \mid l_k \mid l_{k-1}$$

2. Nodos vizinhos ( $n$ ) transmitem informações acerca das localizações dos nodos âncoras que estão em suas respectivas zonas de alcance:

$$a \rightarrow \text{Região de cobertura} \quad \text{HELLO} \mid \text{ID}_a \mid l_k$$

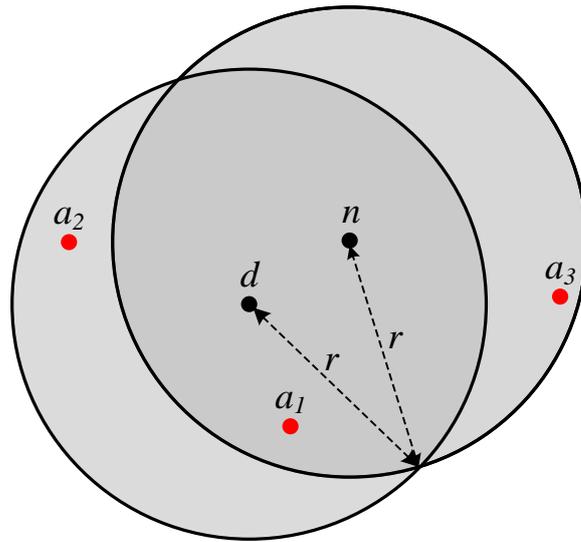
$$n \rightarrow \text{Região de cobertura} \quad \text{HELLO} \mid \text{ID}_n \mid \{\text{ID}_a, l_k\}$$

Segundo os autores de (HU e EVANS, 2004), apesar de ser uma abordagem com maior custo econômico, a grande vantagem da segunda alternativa proposta está no fato de ela também disponibilizar a um nodo desconhecido  $d$  informações a respeito da localização de âncoras que não estão na sua vizinhança direta, mas estão na vizinhança direta dos vizinhos de  $d$ . Tais nodos são classificados como âncoras indiretos (localizados a uma distância de 2 hops do nodo  $d$ ), em contraposição aos chamados âncoras diretos (localizados a uma distância de 1 hop do nodo  $d$ ). Nesse caso, um nodo desconhecido  $d$  sabe que não está a uma distância  $r$  de um âncora indireto, mas pode inferir que está a uma distância máxima  $2r$  desse âncora, uma vez que o mesmo se encontra na vizinhança dos nodos vizinhos a  $d$ .

Âncoras diretos e indiretos não se encaixam, necessariamente, na categorização *insiders* e *outsiders*, respectivamente. Isso acontece porque os critérios que classificam um nodo âncora como sendo *insider* ou *outsider* levam em consideração dois instantes de tempo distintos. Por outro lado, classificar um nodo âncora como direto ou indireto, significa avaliar a relação de vizinhança entre o âncora e o nodo desconhecido apenas no instante de tempo considerado.

A Figura 19 ilustra as condições necessárias para que um nodo âncora seja classificado como direto ou indireto por um nodo desconhecido, de tal forma que essas são as condições utilizadas para filtrar as localizações (amostras) que formam o conjunto  $S'_k$ .

Figura 19 - Tomando o nodo desconhecido  $d$  como referência, os âncoras  $a_1$  e  $a_2$  são considerados âncoras diretos, uma vez que estão na vizinhança de  $d$ . O âncora  $a_3$ , por sua vez, é um âncora indireto para  $d$ , uma vez que está na vizinhança de  $n$ , mas não está na vizinhança de  $d$ . Nesse caso,  $d$  recebe as informações de localização do âncora  $a_3$  por meio de  $n$ .



Fonte: Autor.

Assumindo  $A$  como sendo o conjunto formado por todos os nodos âncoras que estão na vizinhança do nodo desconhecido  $d$  (âncoras diretos), e  $T$  como sendo o conjunto formado por todos os nodos âncoras que estão na vizinhança dos vizinhos de  $d$ , mas não estão na vizinhança de  $d$  (âncoras indiretos), a condição que filtra as amostras em  $S'_k$  é dada pela Equação 13:

$$\text{filter}(s'_k) = \forall a \in A, d(s'_k, l) \leq r \wedge \forall a \in T, r < d(s'_k, l) \leq 2r \quad (13)$$

Dessa forma, todas aquelas amostras que são inconsistentes com as novas observações de possíveis localizações são eliminadas e não compõem o conjunto  $S_k$  que aproxima a PDF posterior  $p(\mathbf{x}_k | Z^k)$ . Àquelas amostras  $s'_k \in S'_k$  que satisfazem as condições do filtro ( $S_k \leftarrow s'_k$ , ou seja,  $s'_k$  passa a ser  $s_k$ ), o algoritmo atribui um peso  $m_k^i = 1$  ( $m_k^i$  passa a ser  $m_k$ ), de tal forma que as coordenadas  $x$  e  $y$  estimadas para o nodo  $d$  no instante de tempo  $k$  são dadas através das Equações 14 e 15:

$$x_k^{est} = \frac{\sum_{i=1}^{\text{size}(S_k)} x_k^i}{\sum_{i=1}^{\text{size}(S_k)} m_k^i} \quad (14)$$

$$y_k^{est} = \frac{\sum_{i=1}^{\text{size}(S_k)} y_k^i}{\sum_{i=1}^{\text{size}(S_k)} m_k^i} \quad (15)$$

onde  $x_k^i$  e  $y_k^i$  são, respectivamente, as coordenadas  $x$  e  $y$  da amostra  $s_k^i$ .

O número  $N$  de amostras que integram o conjunto  $S_k$ , por sua vez, deve ser definido previamente. Assim, considerando que os pesos  $m_k^i$  associados a cada uma das amostras  $s_k^i \in S_k$  têm valor igual a 1, as Equações 14 e 15 apresentadas acima podem ser reescritas conforme as Equações 16 e 17:

$$x_k^{est} = \frac{\sum_{i=1}^N x_k^i}{N} \quad (16)$$

$$y_k^{est} = \frac{\sum_{i=1}^N y_k^i}{N} \quad (17)$$

Diante do exposto na presente seção, observa-se que a grande vantagem da adaptação do método de localização de Monte Carlo para RSSFs móveis reside no fato de que, diferentemente dos demais métodos de localização de nodos apresentados, o algoritmo de Monte Carlo utiliza-se da mobilidade dos nodos para otimizar a precisão das estimativas de localização. Em outras palavras, a alteração topológica da rede gera novas informações de localização a serem utilizadas na etapa de filtragem do algoritmo.

### 3.3.1 Limitações do algoritmo de localização de Monte Carlo para RSSFs móveis

A presente seção tem por objetivo apresentar algumas das limitações do algoritmo de localização de Monte Carlo aplicado a RSSFs móveis. Conforme mencionado durante a apresentação do algoritmo na seção anterior, o mesmo opera sobre um cenário bastante dinâmico, com nodos desconhecidos e nodos âncoras distribuídos de forma irregular e movimentando-se aleatoriamente dentro da área da rede. Considerando o aspecto da mobilidade, esse é o pior cenário possível para a implementação de um algoritmo de localização.

Dessa forma, ao confrontar esse cenário bastante generalista, o método proposto em (HU e EVANS, 2004) encontra obstáculos relacionados, sobretudo, à precisão das estimativas

de localização, aos custos econômicos de implementação da rede e ao consumo de energia por parte dos nodos.

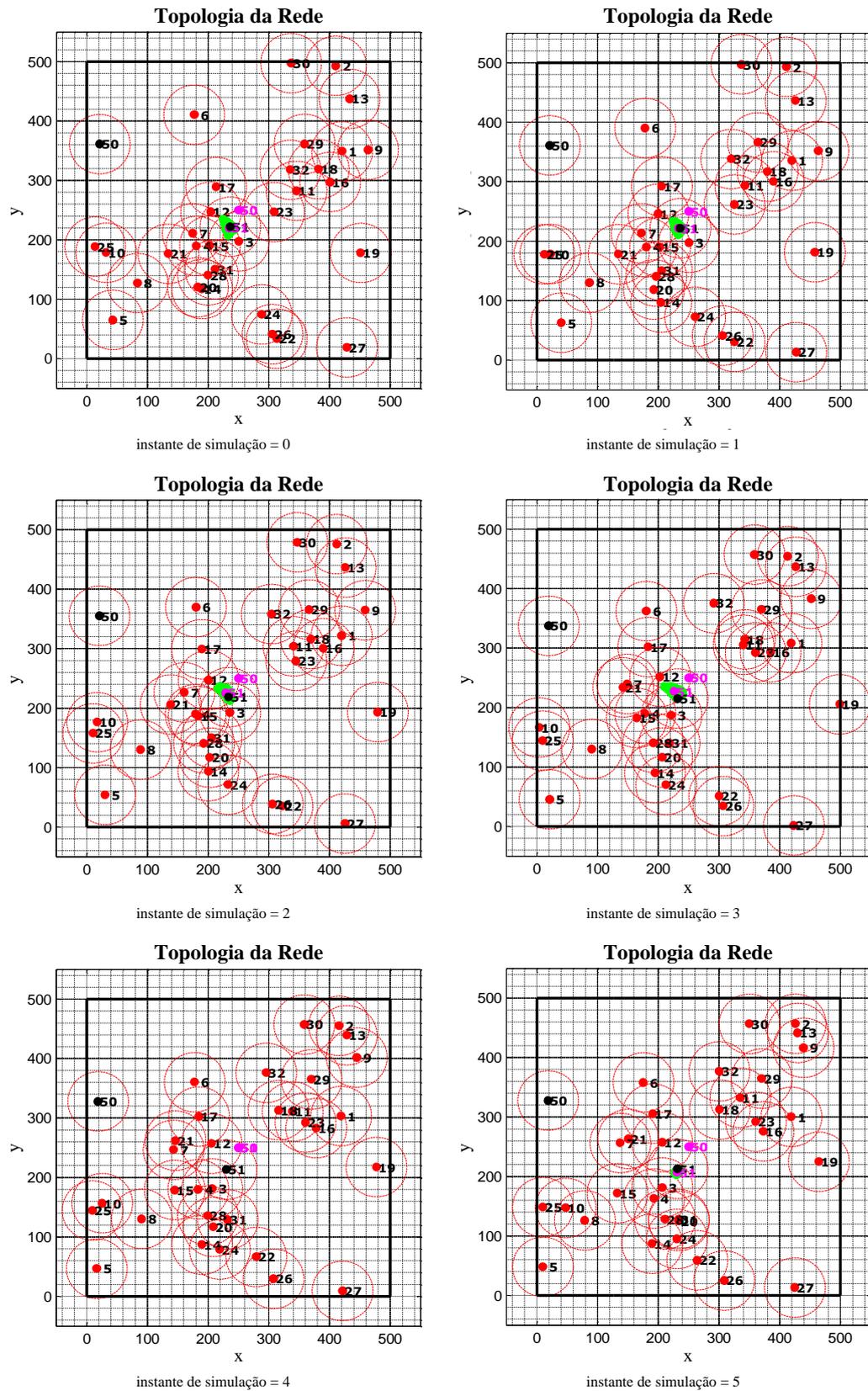
### *3.3.1.1 Etapas de inicialização e de filtragem: dependência de nodos âncoras*

Apesar de o algoritmo de localização de Monte Carlo para RSSFs apresentar uma baixa densidade de nodos âncoras quando comparado a métodos de localização até então existentes, o algoritmo ainda apresenta uma grande dependência em relação à presença de nodos âncoras no cenário. Como exemplo dessa condição, tomam-se as etapas de inicialização e de filtragem do algoritmo.

Conforme já mencionado, durante a etapa de inicialização do método de localização de Monte Carlo, os nodos desconhecidos da rede necessitam estar na zona de alcance de pelo menos um nodo âncora para gerar as amostras do conjunto  $S_0$  e dar início ao processo de localização. Uma vez que o cenário considerado assume a disposição irregular e a movimentação aleatória dos nodos dentro da área da rede, pode-se inferir que, em algumas situações, um nodo desconhecido permaneça uma quantidade de tempo significativa sem estimativas acerca de sua localização. Para muitas aplicações, essa situação é inaceitável.

Da mesma forma, durante a etapa de filtragem do algoritmo, conforme descreve matematicamente a Equação 13, os nodos desconhecidos necessitam receber informações acerca da localização de pelo menos um nodo âncora, seja ele um âncora direto ou indireto. Sem tais informações, os nodos desconhecidos são impossibilitados de eliminar as amostras (localizações) ditas impossíveis que formam o conjunto  $S'_k$  e, conseqüentemente, não são capazes de estimar suas localizações com precisão.

Figura 20 - Influência dos buracos da rede sobre a precisão das estimativas de localização do algoritmo de Monte Carlo para RSSFs móveis.



Fonte: Autor.

Nas duas situações expostas, o método de localização de Monte Carlo posiciona os nodos desconhecidos no centro da rede, o que acaba penalizando o erro de estimativa do algoritmo.

A Figura 20 ilustra graficamente a influência que os chamados buracos da rede (regiões onde não há nodos de referência) exercem sobre a precisão das estimativas de localização do algoritmo de Monte Carlo para RSSFs móveis. A figura mostra os resultados de uma simulação realizada por meio da ferramenta de validação desenvolvida neste trabalho. Os parâmetros de simulação adotados são listados na Tabela 1.

Tabela 1 - Parâmetros de simulação utilizados para avaliar a influência dos chamados buracos da rede sobre o processo de estimativa de localização dos nodos desconhecidos.

PARÂMETROS DE SIMULAÇÃO: INFLUÊNCIA DOS BURACOS DA REDE SOBRE AS ESTIMATIVAS DE LOCALIZAÇÃO	
Área de simulação	500 x 500 unidades de distância
Número total de nodos	320
Densidade de nodos	10
Número de nodos âncoras	32
Densidade de nodos âncoras	1
Velocidade máxima dos nodos	30 unidades de distância/passo de simulação
Alcance de transmissão dos nodos	50 unidades de distância
Número de amostras	50
Modelo de mobilidade	<i>Random Waypoint</i> modificado <sup>9</sup> ( $t_{pausa} = 0$ )
Tempo total de simulação	10 passos de simulação

Fonte: Autor.

Na Figura 20, os pontos em vermelho representam os 32 nodos âncoras existentes na rede (identificados do número 1 ao número 32), os pontos em preto representam 2 nodos desconhecidos da rede (identificados com os números 50 e 51), os pontos em verde indicam as amostras do conjunto  $S_k$  (localizações selecionadas durante a etapa de filtragem do algoritmo), os pontos em lilás indicam as localizações estimadas para os nodos desconhecidos 50 e 51, e as circunferências em vermelho representam as zonas de alcance dos nodos. Como pode ser observado, durante os 6 instantes iniciais de simulação, o nodo desconhecido 50 não entra na zona de alcance de nodos âncoras. Por isso, a localização estimada para esse nodo em cada

<sup>9</sup> O detalhamento do modelo de mobilidade implementado na ferramenta de simulação desenvolvida é feito no Capítulo 5 do presente documento.

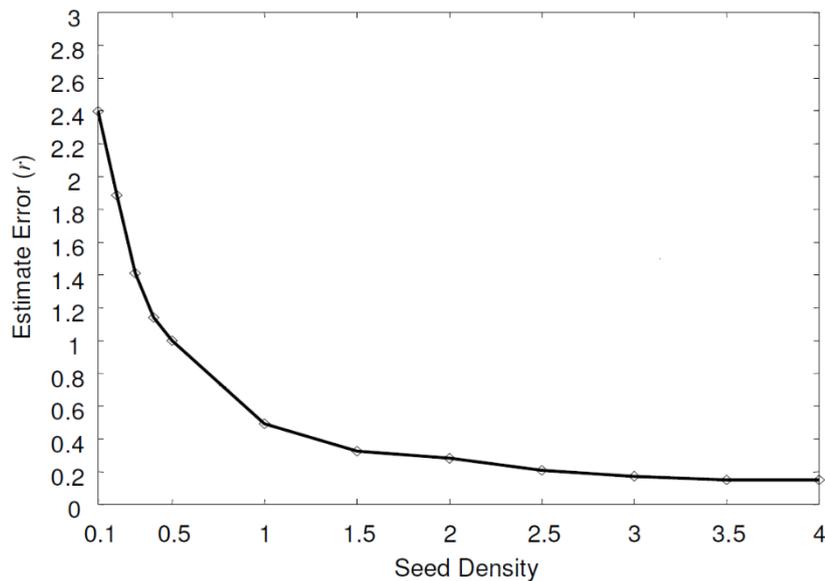
um desses 6 instantes é o ponto central da rede (coordenadas  $x$  e  $y$  iguais a 250), conforme evidencia o ponto em lilás referenciado com a identificação (50) desse nodo. Nesse caso, o erro médio nas estimativas de localização do nodo 50 durante os 6 instantes iniciais da simulação é de 252.55, medido em unidades de distância (normalizado em relação ao raio de alcance  $r$ , o erro é de 5.05).

Já no caso do nodo desconhecido 51, observa-se, no instante de tempo 4, que o mesmo sai da presença de nodos âncoras diretos e indiretos, o que impossibilita a realização da etapa de filtragem do algoritmo para este nodo. Assim, no instante de tempo mencionado, a localização estimada para esse nodo também é o ponto central da rede. Abaixo, listam-se os erros (medidos em unidades de distância) obtidos nas estimativas de localização do nodo 51 para cada um dos instantes iniciais de simulação, bem como o erro médio calculado:

- instante 0: 2.86;
- instante 1: 3.35;
- instante 2: 8.26;
- instante 3: 10.1;
- instante 4: 41.91;
- instante 5: 5.8;
- erro médio para os 6 instantes considerados: 12.05 (normalizado em relação ao raio de alcance  $r$ , o erro é de 0.24).

É possível observar, através da análise dos valores listados, que o erro obtido no instante de tempo 4 exerce um peso bastante significativo sobre o cálculo do erro médio, evidenciando, mais uma vez, o impacto negativo dos buracos da rede sobre a precisão das estimativas de localização do algoritmo de Monte Carlo. O gráfico mostrado na Figura 21 demonstra a influência da densidade de nodos âncoras (*seeds*) da rede sobre o erro médio registrado pelo algoritmo de localização de Monte Carlo. A densidade de nodos âncoras, nesse caso, é definida como sendo a quantidade de nodos âncoras existentes em *1 hop* de transmissão dentro da rede.

Figura 21 - Impacto da densidade de nodos âncoras da rede sobre o erro médio de estimativa do algoritmo de localização de Monte Carlo para RSSFs.



Fonte: (HU e EVANS, 2004).

Obviamente, o aumento da densidade de nodos âncoras da rede torna o processo de localização do algoritmo mais fácil e preciso. No entanto, isso representa um aumento nos custos econômicos de implementação da rede.

Outra análise que pode ser feita tomando como base a densidade de nodos âncoras da rede diz respeito ao valor mínimo da velocidade  $v_{max}$  dos nodos para o qual o algoritmo de Monte Carlo consegue gerar estimativas de localização satisfatórias. Nesse sentido, a velocidade dos nodos da rede impacta no processo de localização do algoritmo de duas maneiras:

1. A redução da velocidade dos nodos torna as estimativas de localização do algoritmo de Monte Carlo mais precisas, uma vez que reduz a área da rede onde estão inseridas as possíveis localizações dos nodos (amostras  $s'_k{}^i$ ,  $i \leq N$ ) na próxima iteração;
2. Por outro lado, a redução da velocidade dos nodos gera uma menor quantidade de novas observações a cada iteração, de tal forma que uma menor quantidade de localizações impossíveis podem ser filtradas.

Assim, considerando a redução da velocidade dos nodos da rede, observa-se que o item 2 é determinante sobre a precisão das estimativas de localização do algoritmo de Monte Carlo, pois mesmo que as amostras  $s'_k{}^i \in S'_k$  sejam geradas com maior precisão durante a etapa de predição, o fato de não haver novas observações (contato com nodos âncoras diretos e/ou

indiretos) faz com que a etapa de filtragem não seja realizada e, dessa forma, a localização estimada para o nodo será o ponto central da rede (o mesmo vale para a etapa de inicialização, caso o nodo desconhecido não esteja em contato com um âncora direto).

Por isso, na medida em que se reduz a velocidade de movimentação dos nodos, o algoritmo de localização de Monte Carlo para RSSFs exige um aumento da densidade de nodos âncoras a fim de gerar estimativas de localização com um valor aceitável para o erro médio. Dessa forma, a localização de nodos através do método de Monte Carlo em RSSFs móveis de baixa velocidade pode mostrar-se inviável, dado o custo elevado de implementação da rede em função da quantidade necessária de nodos âncoras. Uma análise mais detalhada acerca do impacto da velocidade de movimentação dos nodos da rede sobre a precisão das estimativas de localização do algoritmo de Monte Carlo pode ser obtida em (HU e EVANS, 2004).

### 3.3.1.2 Etapa de predição: modelo de mobilidade com pouca informação

Conforme descreve matematicamente a Equação 10, durante a etapa de predição dos métodos de localização instanciados dentro da inferência Bayesiana utiliza-se um modelo de mobilidade para estimar o estado atual do dispositivo alvo na forma de uma PDF preditiva  $p(\mathbf{x}_k | Z^{k-1})$ . Nesse caso, assume-se que o estado atual  $\mathbf{x}_k$  depende unicamente do estado anterior  $\mathbf{x}_{k-1}$  e de uma entrada de controle  $\mathbf{u}_{k-1}$ , de tal forma que o modelo de mobilidade é descrito como a densidade condicional  $p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_{k-1})$ . Assim, quanto maior a quantidade de informações úteis disponibilizadas pelo modelo de mobilidade considerado, maior será a precisão das estimativas de localização dadas pela densidade  $p(\mathbf{x}_k | Z^{k-1})$  e, conseqüentemente (Equação 11), pela PDF posterior  $p(\mathbf{x}_k | Z^k)$ .

No caso do algoritmo de localização de Monte Carlo aplicado a RSSFs móveis, o modelo de mobilidade adotado leva em consideração apenas a velocidade máxima  $v_{max}$  de movimentação dos nodos da rede. Assim, a partir das amostras (possíveis localizações)  $s_{k-1}^i$  obtidas na iteração anterior e da informação sobre sua velocidade máxima, um nodo desconhecido consegue inferir apenas que, no instante de tempo atual ( $k$ ), encontra-se em algum lugar no interior da circunferência de centro  $s_{k-1}^i$  e raio  $v_{max}$ . Nesse sentido, por meio da utilização de um modelo de mobilidade com mais informações úteis acerca da movimentação dos nodos, pode-se de gerar amostras  $s_k^i \in S_k'$  mais precisas, o que contribui para a redução do erro nas estimativas de localização do algoritmo.



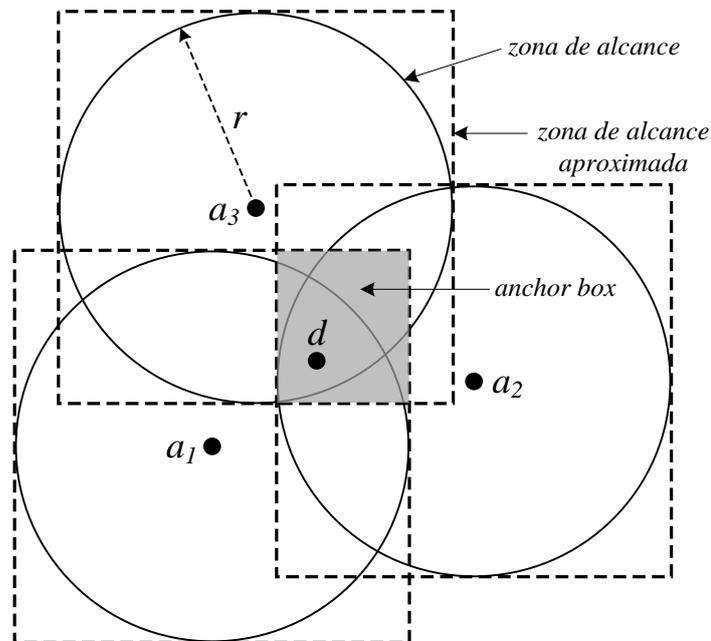
O método usado para delimitar a área (*bounding box*) a partir da qual o algoritmo MCB gera novas amostras funciona da seguinte maneira: um nodo desconhecido  $d$  que recebe informações de localização provenientes de nodos âncoras (distantes 1 ou 2 hops de  $d$ ) constrói uma caixa virtual que cobre a região onde as zonas de alcance de tais âncoras se sobrepõem, conforme ilustra a Figura 23. Em outras palavras, a caixa representa a região da rede onde o nodo desconhecido  $d$  se encontra. A construção da *anchor box*, como é chamada essa região, é dada através das Equações 18 e 19:

$$x_{min} = \max_{j=1}^n (x_j - r) \quad x_{max} = \min_{j=1}^n (x_j + r) \quad (18)$$

$$y_{min} = \max_{j=1}^n (y_j - r) \quad y_{max} = \min_{j=1}^n (y_j + r) \quad (19)$$

onde  $x_j$  e  $y_j$  são as coordenadas do correspondente nodo âncora  $j$ ,  $n$  é o número de nodos âncoras que fornecem informações ao nodo  $d$  e  $r$  é o raio de alcance dos nodos âncoras da rede. Caso o âncora  $j$  esteja a uma distância de 2 hops de  $d$ , o termo  $r$  é substituído por  $2r$  nas Equações acima.

Figura 23 - Construção da *anchor box*.



Fonte: Adaptação de (BAGGIO e LANGENDOEN, 2008).

Uma *bounding box*, portanto, representa a forma computacional mais simples de se determinar os limites lineares de determinada região, sendo utilizada em inúmeras aplicações geométricas, como na detecção de objetos escondidos e nos métodos empregados para evitar colisões. Para o caso específico mencionado, a *bounding box* é chamada de *anchor box* pelo fato de ser construída a partir das informações de localização de nodos âncoras.

A estimativa da localização do nodo  $d$  utilizando somente a *anchor box* ocorre apenas para o caso onde o conjunto de amostras  $S_{k-1}$  correspondente ao nodo  $d$  está vazio, como no instante de inicialização. No entanto, para a situação onde o nodo  $d$  possui amostras geradas na iteração anterior (instante de tempo  $k-1$ ), a *bounding box* passa a ser construída com informações adicionais: para cada uma das amostras  $s_{k-1}^i$  que formam o conjunto  $S_{k-1}$ , constrói-se uma caixa adicional de tamanho  $2v_{max}$  e centro no ponto indicado por  $s_{k-1}^i$ , conforme Equações 20 e 21:

$$x_{min}^i = \max(x_{min}, x_{k-1}^i - v_{max}) \quad x_{max}^i = \min(x_{max}, x_{k-1}^i + v_{max}) \quad (20)$$

$$y_{min}^i = \max(y_{min}, y_{k-1}^i - v_{max}) \quad y_{max}^i = \min(y_{max}, y_{k-1}^i + v_{max}) \quad (21)$$

onde  $x_{k-1}^i$  e  $y_{k-1}^i$  são as coordenadas da amostra  $s_{k-1}^i$ . Essa caixa virtual atualizada, chamada *sample box*, delimita, para cada uma das amostras do conjunto  $S_{k-1}$ , a área através da qual o nodo  $d$  pode se mover durante um intervalo de tempo. Dessa forma, todas as amostras que integram o conjunto  $S_k'$  do correspondente nodo desconhecido estão contidas na *sample box*, que representa a área sobre a qual a função de amostragem atua. Caso o nodo  $d$  não receba informações provenientes de nodos âncoras, a *sample box* é obtida somente através das informações provenientes das amostras  $s_{k-1}^i$  e da velocidade máxima do nodo. A construção da *sample box*, portanto, corresponde à etapa de predição do algoritmo MCB. A etapa de filtragem, por sua vez, é realizada da mesma forma como ocorre no algoritmo MCL original, conforme Equação 13.

A principal vantagem do algoritmo MCB em relação ao método descrito em (HU e EVANS, 2004) está na redução do tempo de computação necessário para gerar as amostras utilizadas no processo de localização dos nodos, uma vez que, através da construção da *sample box*, o primeiro método citado procura antecipar alguma avaliação da informação para a função de amostragem.

Já em (STEVENS-NAVARRO, VIVEKANANDAN e WONG, 2007), Navarro et. al apresentam dois novos algoritmos de localização para RSSFs móveis também construídos sobre o método MCL original. A principal ideia por trás do primeiro algoritmo apresentado, chamado *Dual MCL*, está na inversão do processo de amostragem das densidades de probabilidade  $p(\mathbf{x}_k | Z^{k-1})$  e  $p(\mathbf{x}_k | Z^k)$  utilizadas no método MCL original. Dessa forma, a etapa de predição do algoritmo *Dual MCL* gera as amostras do conjunto  $S'_k$  através das informações de localização provenientes de nodos âncoras vizinhos (distantes 1 ou 2 *hops* do nodo desconhecido considerado), enquanto a etapa de atualização filtra as amostras de  $S'_k$  por meio das informações disponibilizadas pelo modelo de mobilidade adotado para os nodos. A condição que gera as amostras durante a etapa de predição do algoritmo *Dual MCL* é a mesma descrita na Equação 10. Da mesma forma, a condição que filtra as amostras do conjunto  $S'_k$  durante a etapa de atualização considera apenas a velocidade máxima  $v_{max}$  dos nodos, conforme Equação 22:

$$filter\_dual(s'_k{}^i) = d(s'_k{}^i, s_{k-1}^i) \leq v_{max} \cdot \Delta k \quad (22)$$

onde  $\Delta k$  é o intervalo de tempo entre duas interações consecutivas (*time step*).

No segundo método de localização apresentado em (STEVENS-NAVARRO, VIVEKANANDAN e WONG, 2007), referenciado como *Mixture MCL*, as amostras são geradas através do algoritmo MCL original e do algoritmo *Dual MCL*. Um parâmetro  $\emptyset$  é definido como  $0 \leq \emptyset \leq 1$ , de tal forma que o algoritmo *Mixture MCL* gera amostras com probabilidade  $1 - \emptyset$  através do processo de amostragem do método MCL original, ao passo que amostras com probabilidade  $\emptyset$  são geradas através da metodologia de amostragem utilizada no método *Dual MCL*.

Em comparação com o método MCL original, tanto o algoritmo *Dual MCL* quanto o algoritmo *Mixture MCL* apresentam uma maior precisão nas estimativas de localização. No que diz respeito ao compromisso entre o tempo de computação e a precisão nas estimativas de localização, o algoritmo *Mixture MCL* mostra um desempenho superior ao verificado com os algoritmos MCL original e *Dual MCL*.

Em (MARTINS, CHEN e SEZAKI, 2009), Martins et. al também apresentam uma variação do método de localização proposto por Hu e Evans. A grande contribuição do algoritmo OTMCL (do inglês *Oriented Tracking-based Monte Carlo Localization*), como é cha-

mado, está na inclusão da informação de orientação (direção e sentido) ao modelo de mobilidade considerado pela função de amostragem na etapa de predição.

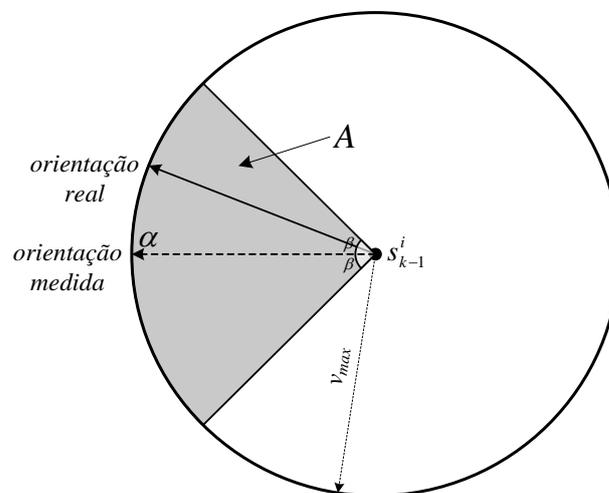
Durante a etapa de inicialização, assim como em (BAGGIO e LANGENDOEN, 2008), o algoritmo OTMCL define uma área inicial de predição que consiste, na verdade, em uma *anchor box* cujas medidas são obtidas através das Equações 18 e 19. Considerando que, no instante de tempo inicial  $k = 0$ , a equação de transição  $p(\mathbf{x}_0 | Z^{k-1})$  não é definida ( $Z^{k-1} = \emptyset$ ), as amostras  $s_{\theta}^i$  são extraídas diretamente da respectiva *anchor box* e a localização inicial do nodo é definida como sendo a média das amostras coletadas. Após a etapa de inicialização, as estimativas de localização são obtidas através das etapas de predição e de filtragem.

Na etapa de predição, as amostras  $s_k^i$  são selecionadas de acordo com a equação de transição do algoritmo OTMCL, a qual é definida pela PDF preditiva  $p(\mathbf{x}_k | Z^{k-1})$  mostrada na Equação 23:

$$p(\mathbf{x}_k | Z^{k-1}) = \begin{cases} \frac{1}{A} & \text{se } d(s_k^i, s_{k-1}^i) < v_{max} \\ 0 & \text{se } d(s_k^i, s_{k-1}^i) \geq v_{max} \end{cases} \quad (23)$$

onde  $A$  indica a área ilustrada na Figura 24 e representada algebricamente através da Equação 24, a qual é escrita em coordenadas polares. O valor de  $\alpha$  conforme mostrado na Figura 24 pode ser obtido através de dispositivos especializados, como acelerômetros e giroscópios. O termo  $\beta$  refere-se ao erro de medição associado ao sensor utilizado.

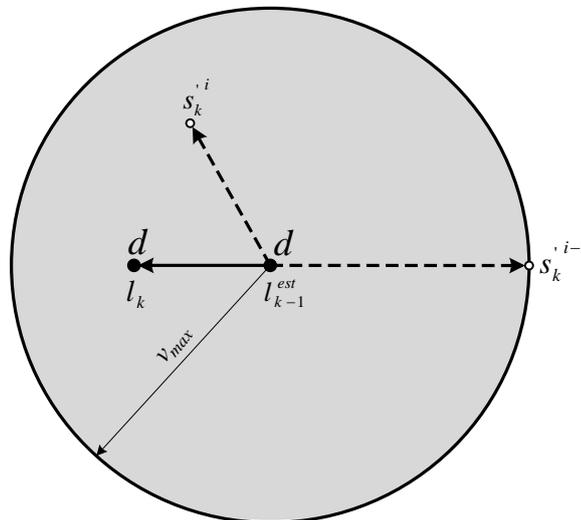
Figura 24 - Setor que contém as amostras do conjunto  $S_k^i$ .



$$A = \frac{1}{2} \int_{\alpha-\beta}^{\alpha+\beta} v_{max}^2 d\theta \quad (24)$$

Em comparação ao método MCL original, a inclusão da medição de orientação à equação de transição do algoritmo OTMCL reduz o número de amostras do conjunto  $S'_k$  que são eliminadas durante a etapa de filtragem, o que representa uma redução no tempo de processamento necessário para gerar amostras de localização válidas (conjunto  $S_k$ ). Isso pode ser visualizado através da análise das Equações 12 e 23, que evidenciam o fato de que a área de predição é inversamente proporcional à probabilidade de se obter uma amostra válida. Graficamente, a Figura 25 também demonstra essa condição, ilustrando a contribuição negativa das amostras  $s_k^{i-1}$  e  $s_k^i$  ao processo de localização quando se considera apenas a velocidade máxima dos nodos.

Figura 25 - Contribuição negativa das amostras  $s_k^{i-1}$  e  $s_k^i$  ao processo de localização do algoritmo de Monte Carlo quando se considera apenas a velocidade máxima dos nodos. No caso ilustrado,  $l_{k-1}^{est}$  corresponde à localização estimada do nodo  $d$  no instante de tempo  $k-1$  e  $l_k$  corresponde à localização real do nodo  $d$  no instante de tempo  $k$ .



Fonte: Adaptação de (MARTINS, CHEN e SEZAKI, 2009).

Caso o algoritmo OTMCL não consiga validar um número suficiente de amostras geradas a partir da Equação 23, as amostras  $s_k^i$  são selecionadas diretamente de uma *anchor box* definida pelas Equações 18 e 19 e filtradas de acordo com as condições definidas pela Equação 13. Na ausência de nodos âncoras, o algoritmo utiliza a medição de orientação  $\alpha_k$  do ins-

tante atual e seleciona uma velocidade aleatória dentro do intervalo  $[0, v_{max}]$  para estimar a atual localização do nodo desconhecido. Esse procedimento contribui para uma redução no erro das estimativas de localização do algoritmo OTMCL quando comparado aos algoritmos MCB e MCL original, visto que, nessa situação, os dois últimos algoritmos mencionados estimam a posição do nodo no ponto central da área da rede.

Por fim, no trabalho apresentado em (DIL, DULMAN e HAVINGA, 2006), Dil et. al propõem a integração de medições *range-based* ao método *range-free* de localização proposto por Hu e Evans, objetivando reduzir os erros das estimativas de localização do método MCL original. Com isso, os autores lidam com a relação de compromisso entre os erros de medição de distância e os custos computacionais do processamento da informação. Da mesma forma como ocorre em (BAGGIO e LANGENDOEN, 2008) e (MARTINS, CHEN e SEZAKI, 2009), o algoritmo proposto em (DIL, DULMAN e HAVINGA, 2006) procura antecipar alguma avaliação da informação para a função de amostragem com o intuito de reduzir os custos computacionais associados às diversas iterações da etapa de filtragem do algoritmo MCL. Dil et. al consideram o mesmo modelo de mobilidade empregado em (HU e EVANS, 2004) e as medições de distância são realizadas através do método RSSI, conforme descrito na Seção 2.2.2.4. De acordo com (OLIVEIRA, 2009), em comparação ao método MCL original, o algoritmo em questão apresenta custos computacionais menos vulneráveis a variações paramétricas do algoritmo e da rede bem como um desempenho menos dependente da velocidade do nó. Isso, no entanto, pode se tornar uma desvantagem quando se leva em consideração diferentes padrões de movimentação (CAMP, BOLENG e DAVIES, 2002).

### 3.4 RESUMO DO CAPÍTULO

No presente capítulo foram introduzidos aspectos teóricos acerca dos problemas instanciados dentro da chamada Inferência Bayesiana, tendo como objetivo principal analisar o problema da localização de nodos em RSSFs segundo essa lógica. Apresenta-se, então, como alternativa de solução para o problema de localização mencionado, o método de localização de Monte Carlo (MCL), definindo-o, inicialmente, sobre o campo da robótica móvel (DELLAERT, FOX, *et al.*, 1999) e, em seguida, explicando de maneira detalhada o funcionamento do algoritmo de localização de Monte Carlo para RSSFs móveis proposto por Hu e Evans em (HU e EVANS, 2004). Esse é o método de localização sobre o qual se constrói a proposta desta dissertação e, por isso, o presente capítulo ainda destina algumas de suas se-

ções ao apontamento das limitações do algoritmo mencionado bem como à apresentação de trabalhos científicos relacionados ao mesmo.



## 4 ESQUEMA DE LOCALIZAÇÃO PROPOSTO

Assim como os trabalhos apresentados na Seção 3.3.2 do presente documento, o esquema de localização de nodos em RSSFs móveis proposto nesta dissertação também é construído sobre o método de localização de Monte Carlo descrito em (HU e EVANS, 2004). Basicamente, a proposta reúne elementos explorados em (HU e EVANS, 2004) (Seção 3.3), (HSIEH e WANG, 2006) (Seção 2.2.3.2), (BAGGIO e LANGENDOEN, 2008) (Seção 3.3.2) e (MARTINS, CHEN e SEZAKI, 2009) (Seção 3.3.2).

Como principais contribuições do esquema proposto, destacam-se:

- a utilização da técnica de localização *range-free* baseada na contagem de saltos (NICULESCU e NATH, 2001) (HSIEH e WANG, 2006) para auxiliar o processo de localização durante a etapa de inicialização do algoritmo de Monte Carlo;
- a utilização de medições de velocidade e de orientação dos nodos para realizar o processo de amostragem durante a etapa de predição do algoritmo.

Logo, trata-se de uma abordagem totalmente *range-free* que objetiva trabalhar as limitações inerentes ao método MCL original, conforme apresentado na Seção 3.3.1.

O presente capítulo visa explicar o funcionamento do método de localização proposto, detalhando cada uma das etapas que o integram.

### 4.1 ETAPA DE INICIALIZAÇÃO

A etapa de inicialização dos esquemas de localização para RSSFs baseados no método de Monte Carlo contribui para um aumento do erro médio nas estimativas de localização do algoritmo considerado (obviamente, quando os instantes iniciais do processo de estimativa são considerados no cálculo do erro). Isso ocorre, principalmente, porque a primeira estimativa de localização de um nodo desconhecido através do método de Monte Carlo exige que esse nodo esteja a 1 (HU e EVANS, 2004) ou, no máximo, 2 *hops* (BAGGIO e LANGENDOEN, 2008) (MARTINS, CHEN e SEZAKI, 2009) de distância de pelo menos um nodo âncora. Nesse caso, dada a disposição irregular e a movimentação aleatória dos nodos dentro do cenário considerado, em algumas situações é possível que um nodo desconhecido permaneça um período de tempo significativo sem estimativas acerca de sua localização, o que, para muitas aplicações, é inaceitável. A Figura 20 (Seção 3.3.1) ilustra claramente essa condição: durante os 6 instantes iniciais da simulação apresentada, o nodo desconhecido identificado pelo núme-

ro 50 não entra na zona de alcance de nodos de referência e, por isso, sua localização é estimada no ponto central da área da rede.

Diante disso, o esquema de localização proposto nesta dissertação sugere a utilização de uma abordagem *range-free* baseada na contagem de saltos durante a etapa de inicialização do algoritmo. Conforme exposto na Seção 2.2.3.2, técnicas de localização para RSSFs baseadas na contagem de saltos valem-se da disseminação de mensagens iniciadas por nodos âncoras através da rede, e, por isso, não exigem que os nodos desconhecidos estejam na zona de alcance de nodos de referência para a obtenção de estimativas de localização, bem como não necessitam de uma densidade de nodos âncoras elevada para o seu funcionamento. Por isso, a técnica de *hop counting*, como também é chamada, apresenta-se como uma possível solução para o problema descrito no parágrafo anterior.

Em (HSIEH e WANG, 2006), conforme também apresentado na Seção 2.2.3.2, propõe-se uma alternativa eficaz para a utilização da técnica de contagem de saltos (algoritmo DV-HOP) em RSSFs móveis. Até então, a utilização dessa técnica em cenários com nodos móveis mostrava-se inviável, principalmente pelo fato de as alterações topológicas da rede tornarem desatualizadas as mensagens propagadas pelos nodos âncoras através de longas distâncias. Diante disso, por meio de uma abordagem que implementa o algoritmo DV-HOP localmente, Hsieh e Wang (HSIEH e WANG, 2006) eliminam esse problema e, ao mesmo tempo, obtêm um aumento na precisão das estimativas de localização em relação ao método DV-HOP original, bem como uma redução no fluxo de mensagens e no consumo de energia da rede. Por isso, durante a etapa de inicialização do esquema de localização proposto, opta-se pela implementação de uma técnica de contagem de saltos que opere localmente dentro da área da rede.

Nesse caso, no instante inicial do processo de localização (instante de tempo  $k = 0$ ), todos os nodos âncoras da rede propagam uma mensagem identificada contendo suas respectivas coordenadas e o parâmetro  $fh$  (do inglês *flooding-hop*) (HSIEH e WANG, 2006), o qual indica a distância, em *hops*, através da qual a mensagem deve se propagar. O valor inicial desse parâmetro deve ser previamente definido para cada aplicação, de tal forma que seu valor depende, sobretudo, da densidade de nodos e da densidade de nodos âncoras da rede considerada. A cada *hop* percorrido pela mensagem, o parâmetro  $fh$  é decrementado, fazendo com que a propagação cesse quando  $fh = 0$ . Da mesma forma como ocorre para os demais algoritmos de localização que se utilizam da contagem de saltos, os nodos da rede irão receber mensagens com valores distintos de  $fh$  associados a um único nodo âncora. Nesse caso, os nodos devem decrementar e retransmitir apenas aqueles valores de  $fh$  que são superiores aos respec-

tivos valores armazenados. Observa-se que essa lógica é inversa à adotada pelo método DV-HOP original, uma vez que o mesmo trabalha com o incremento da contagem de *hops*.

Conforme mencionado no parágrafo inicial desta seção, durante a etapa de inicialização dos algoritmos mostrados em (HU e EVANS, 2004), (BAGGIO e LANGENDOEN, 2008) e (MARTINS, CHEN e SEZAKI, 2009), os nodos desconhecidos só são capazes de estimar suas coordenadas por meio da obtenção de informações provenientes de âncoras situados a 1 ou 2 *hops* de distância. Assim, no caso dos métodos explorados em (BAGGIO e LANGENDOEN, 2008) e (MARTINS, CHEN e SEZAKI, 2009), um nodo desconhecido  $d$  que recebe informações diretamente de nodos âncoras, é capaz de construir uma *anchor box* através das Equações 18 e 19. Caso as informações relacionadas aos nodos âncoras sejam repassadas a  $d$  através de um nodo desconhecido vizinho, a *anchor box* é construída substituindo o termo  $r$  (1 *hop*) por  $2r$  (2 *hops*) nas Equações 18 e 19.

Seguindo essa mesma lógica, através do método de contagem de saltos proposto, objetiva-se aumentar o alcance das mensagens propagadas pelos nodos âncoras, de tal forma que, obtido o valor  $fh_j$  associado ao âncora  $j$ , um nodo desconhecido  $d$  seja capaz de calcular a distância mínima  $hd_j$  (do inglês *hop-distance*), em *hops*, em relação a  $j$ , através da Equação 25:

$$hd_j = fhi - fh_j \quad (25)$$

onde  $fhi$  representa o valor inicial do *flooding-hop*, ou seja, a distância em *hops* que a mensagem iniciada por cada nodo âncora deve percorrer. Seu valor, conhecido por todos os nodos desconhecidos, é o mesmo para todos os âncoras da rede.

Finalmente, em posse das coordenadas  $x_j$  e  $y_j$  e do valor  $hd_j$  associados ao âncora  $j$ , o nodo desconhecido  $d$  é capaz de construir uma *anchor box* cujos limites são obtidos através das Equações 26 e 27:

$$x_{min} = \max_{j=1}^n (x_j - hd_j \cdot r) \quad x_{max} = \min_{j=1}^n (x_j + hd_j \cdot r) \quad (26)$$

$$y_{min} = \max_{j=1}^n (y_j - hd_j \cdot r) \quad y_{max} = \min_{j=1}^n (y_j + hd_j \cdot r) \quad (27)$$

onde  $n$  é o número de nodos âncoras cujas mensagens alcançam o nodo desconhecido  $d$  e  $r$  é o raio de alcance dos nodos âncoras da rede. Caso um dos parâmetros obtidos por meio das Equações 26 e 27 ( $x_{min}$ ,  $x_{max}$ ,  $y_{min}$  e/ou  $y_{max}$ ) ultrapasse o limite correspondente da área da rede,

o algoritmo responsável pela construção da *anchor box* atribui ao parâmetro calculado o valor associado ao limite excedido.

Assim, definido o número  $N$  de amostras necessárias para estimar a localização dos nodos desconhecidos, a etapa de inicialização do esquema proposto é realizada conforme pseudocódigo comentado no Algoritmo 2.

---

Algoritmo 2 - Pseudocódigo comentado da etapa de inicialização do esquema de localização proposto.

---

Legenda:

$fh_j^{rec}$  – *flooding-hop* (referente ao âncora  $j$ ) recebido pelo nodo  $n$

$fh_n^j$  – *flooding-hop* (referente ao âncora  $j$ ) armazenado pelo nodo  $n$

1. Cada nodo âncora  $j$  da rede propaga uma mensagem identificada para sua respectiva zona de alcance. Essa mensagem contém a localização  $l_j = (x_j, y_j)$  do respectivo nodo âncora e o valor inicial ( $fhi$ ) atribuído ao *flooding-hop*.

$j \rightarrow$  zona de alcance                      HELLO | ID $_j$  |  $l_j$  |  $fhi$

2. Cada nodo  $n$  da rede (âncoras e desconhecidos) que recebe uma mensagem iniciada por um âncora decrementa o valor do  $fh$  recebido e, caso  $fh_j^{rec} > fh_n^j$ , repassa a mensagem à sua respectiva zona de alcance até que  $fh = 0$ .

$n \leftarrow$  zona de alcance                      HELLO | ID $_j$  |  $l_j$  |  $fh_j^{rec}$

$fh_j^{rec} = fh_j^{rec} - 1$

**if**  $fh_j^{rec} > fh_n^j$

$fh_n^j = fh_j^{rec}$ ;

$n \rightarrow$  zona de alcance                      HELLO | ID $_j$  |  $l_j$  |  $fh_n^j$

**end**

### 3. Estimativa das localizações dos nodos desconhecidos

3.1. Uma vez que todas as mensagens iniciadas pelos âncoras percorreram a distância, em *hops*, especificada pelo parâmetro  $fhi$ , todos os nodos desconhecidos da rede que possuem informações provenientes de nodos âncoras estimam suas respectivas localizações por meio do processo apresentado abaixo.

#### 3.1.1. Cálculo das distâncias $hd_j$

---

---


$$hd_j = fhi - fh_j \quad (25)$$

### 3.1.2. Construção da anchor box

$$x_{min} = \max_{j=1}^n (x_j - hd_j \cdot r) \quad x_{max} = \min_{j=1}^n (x_j + hd_j \cdot r) \quad (26)$$

$$y_{min} = \max_{j=1}^n (y_j - hd_j \cdot r) \quad y_{max} = \min_{j=1}^n (y_j + hd_j \cdot r) \quad (27)$$

### 3.1.3. Construção do conjunto $S_0$ de amostras

**for**  $i = 1:N$

$$x_0^i \leftarrow \text{rand}[x_{min}, x_{max}];$$

$$y_0^i \leftarrow \text{rand}[y_{min}, y_{max}];$$

$$s_0^i \leftarrow (x_0^i, y_0^i);$$

$$S_0 \leftarrow s_0^i;$$

**end**

### 3.1.4. Estimativa da localização

$$x_0^{est} = \frac{\sum_{i=1}^N x_0^i}{N};$$

$$y_0^{est} = \frac{\sum_{i=1}^N y_0^i}{N};$$

$$l_0^{est} \leftarrow (x_0^{est}, y_0^{est});$$

**3.2.** Os nodos desconhecidos que não receberam informações iniciadas por nodos âncoras terão suas coordenadas estimadas no ponto central da área da rede.

$$x_0^{est} = \frac{x_{range}}{2}, \text{ onde } x_{range} \text{ é a dimensão da rede no eixo } x;$$

$$y_0^{est} = \frac{y_{range}}{2}, \text{ onde } y_{range} \text{ é a dimensão da rede no eixo } y;$$


---

Por meio das mudanças sugeridas na forma como são realizadas as estimativas de localização durante a etapa de inicialização dos algoritmos baseados no método MCL, objetiva-se:

- a redução do erro médio das estimativas de localização durante a etapa de inicialização;

- a redução do número de nodos desconhecidos que não são capazes de estimar, mesmo que com precisão limitada, suas respectivas localizações durante a etapa de inicialização;
- a redução da dependência de nodos âncoras para a realização das estimativas de localização.

## 4.2 ETAPAS DE PREDIÇÃO E DE FILTRAGEM

Conforme exposto na Seção 3.2 da presente dissertação, durante a etapa de predição do método de localização de Monte Carlo são obtidas as amostras  $s_k^i \in S_k'$  referentes ao instante de tempo  $k$  por meio da equação de transição  $p(\mathbf{x}_k | Z^{k-1})$  definida na Equação 10, onde a densidade de probabilidade condicional  $p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_{k-1})$  descreve a aplicação de um modelo de mobilidade sobre todas as amostras  $s_{k-1}^i \in S_{k-1}$  referentes ao instante de tempo anterior  $k-1$ . Cada uma das amostras  $s_k^i$ , então, é filtrada de acordo com as condições definidas na Equação 13, de tal forma que o conjunto  $S_k$  é formado somente por aquelas amostras que satisfazem as condições do filtro.

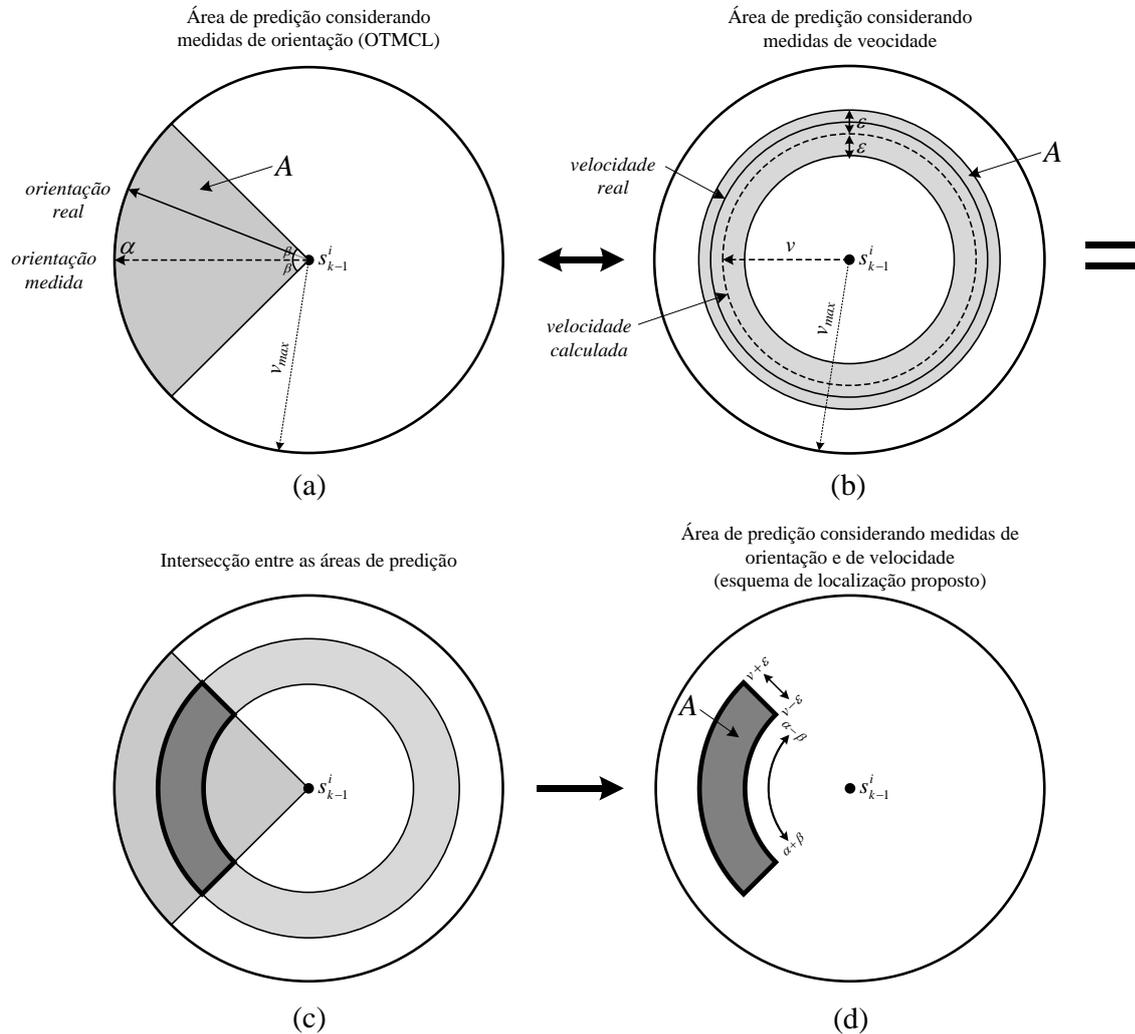
Dessa forma, pode-se afirmar que a quantidade e a qualidade das informações acerca do modelo de mobilidade considerado na etapa de predição influenciam diretamente na precisão das estimativas de localização e no tempo de computação necessário para a obtenção de amostras válidas.

No algoritmo de localização proposto por Hu e Evans, por exemplo, o modelo de mobilidade considerado pela função de amostragem dispõe apenas da velocidade máxima dos nodos da rede, de tal forma que a equação de transição de estado, nesse caso, é definida pela Equação 12. Portanto, as amostras que integram o conjunto  $S_k'$  são geradas aleatoriamente dentro da área compreendida pelas circunferências de raio  $v_{max}$  com centro nas amostras  $s_{k-1}^i$ . A utilização de uma área de predição construída apenas com base na velocidade máxima dos nodos contribui significativamente para o aumento do tempo de computação necessário para a obtenção de  $N$  amostras  $s_k^i$  válidas, conforme ilustra a Figura 25. No caso do algoritmo apresentado em (HU e EVANS, 2004), Hu e Evans adotam  $N = 50$  (número de amostras) e definem um número máximo de 400 tentativas por amostra do conjunto  $S_{k-1}$ , o que significa que o algoritmo pode repetir até 20 mil vezes as etapas de predição e de filtragem antes de obter as  $N$  amostras válidas.

Levando isso em consideração, em (MARTINS, CHEN e SEZAKI, 2009), Martins et. al propõem a inclusão de medidas de orientação do movimento dos nodos ao modelo de mobilidade utilizado pela função de amostragem durante a etapa de predição do algoritmo proposto por Hu e Evans. De acordo com a ilustração mostrada na Figura 24, a construção da área de predição do algoritmo OTMCL, como é chamado, leva em consideração a velocidade máxima  $v_{max}$  dos nodos da rede, a medida de orientação  $\alpha$  fornecida pelo sensor no instante de tempo atual e a margem de erro  $\beta$  associada às medições do sensor considerado. Assim, a equação de transição de estado usada no processo de amostragem do algoritmo OTMCL é definida por meio da Equação 23, onde a área  $A$  corresponde a um setor circular da circunferência de raio  $v_{max}$  e centro na amostra  $s_{k-1}^i$ , conforme Equação 24. Portanto, considerando que a probabilidade de se obter uma amostra válida é inversamente proporcional ao tamanho da área de predição adotada (análise das Equações 12 e 23), pode-se afirmar que a equação de transição definida pela Equação 23 (MARTINS, CHEN e SEZAKI, 2009) aumenta a probabilidade de obtenção de amostras válidas quando comparada à Equação 12 (HU e EVANS, 2004), já que a área de predição adotada pela primeira corresponde a uma parcela da área considerada pela segunda.

Seguindo essa mesma lógica, o esquema de localização aqui sugerido propõe a inclusão de medições da velocidade de movimentação dos nodos ao modelo de mobilidade considerado pelo algoritmo OTMCL. Dessa forma, a área de predição sobre a qual atua a função de amostragem do método de localização proposto é definida conforme mostra a Figura 26(d), onde  $v$  corresponde à média das velocidades desenvolvidas pelo nodo durante o intervalo de tempo  $k - (k-1)$  e  $\varepsilon$  refere-se ao erro de medição associado ao sensor de velocidade considerado. Vale lembrar, neste ponto, que os valores de velocidade medidos representam as distâncias percorridas pelo nodo dentro do intervalo de simulação, conforme mencionado na Seção 3.3.

Figura 26 - Definição da área de predição do esquema de localização proposto.



Fonte: Autor.

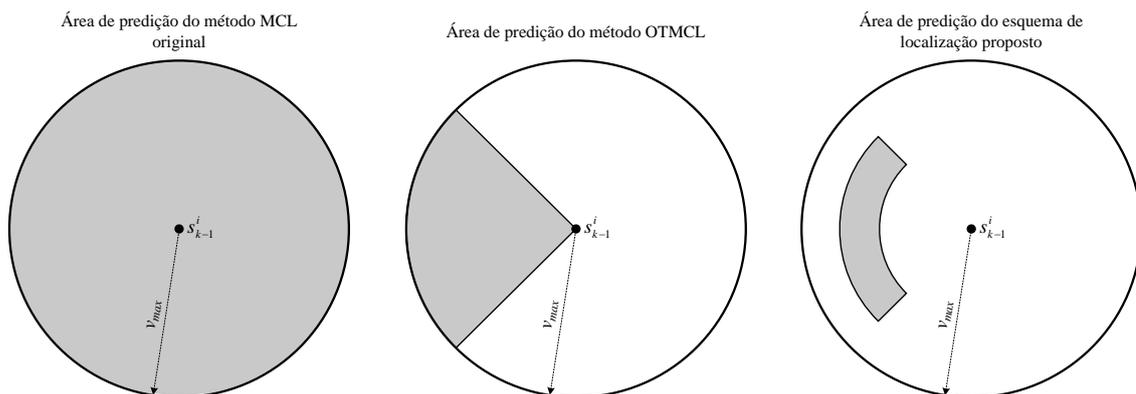
Partindo da Equação 24, que determina a área do setor circular ilustrado na Figura 26(a), chega-se à Equação 28, escrita em coordenadas cilíndricas, que define a área mostrada na Figura 26(d).

$$A = \frac{1}{2} \int_{\alpha-\beta}^{\alpha+\beta} \int_{v-\varepsilon}^{v+\varepsilon} v^2 dv d\theta \quad (28)$$

Caso um dos limites de integração relacionados à velocidade do nodo ultrapasse o limite real correspondente ( $v - \varepsilon < 0$  e/ou  $v + \varepsilon > v_{max}$ ), o algoritmo atribui a tal limite de integração o valor associado ao limite real excedido ( $v - \varepsilon = 0$  e/ou  $v + \varepsilon = v_{max}$ ).

A equação de transição do esquema de localização proposto, por sua vez, é dada através da PDF preditiva  $p(\mathbf{x}_k | Z^{k-1})$  definida na Equação 23, assumindo, no entanto, uma área  $A$  dada pela Equação 28 e o uso da média  $v$  das velocidades medidas no lugar do parâmetro  $v_{max}$ . Portanto, novamente considerando a relação inversa de proporcionalidade entre o tamanho da área de predição e a probabilidade de obtenção de uma amostra válida, espera-se um aumento na precisão das estimativas de localização e uma redução no tempo de computação das etapas de predição e de filtragem do método proposto em comparação aos métodos MCL, MCB e OTMCL, tendo em vista a redução da área de predição do primeiro em relação aos demais. Para fins comparativos, a Figura 27 ilustra as áreas de predição utilizadas pelos métodos MCL original, OTMCL e pelo esquema de localização proposto nesta seção.

Figura 27 - Áreas de predição sobre as quais atuam as funções de amostragem do método MCL original, do método OTMCL e do esquema de localização proposto.



Fonte: Autor.

O modelo de mobilidade associado aos nodos da rede durante as simulações dos algoritmos MCL original, MCB, OTMCL, bem como do esquema de localização proposto, trata-se de um modelo de mobilidade randômica (*Random Waypoint*) modificado, conforme será detalhado no Capítulo 5. Diferentemente do modelo *Random Waypoint* tradicional (JOHNSON e MALTZ, 1996), o qual define uma única velocidade para cada intervalo de movimentação do nodo, o modelo modificado referido assume que os nodos podem variar aleatoriamente suas velocidades dentro do intervalo  $[0, v_{max}]$  durante cada intervalo de movimentação. Nesse caso, durante o tempo de simulação considerado, alterações na velocidade de movimentação dos nodos podem ocorrer em uma frequência maior do que alterações na orientação de suas trajetórias. Por isso, o esquema proposto trabalha com a média das velocidades verificadas entre dois instantes de simulação. O número de medições de velocidade obtido

durante esse intervalo de tempo está relacionado ao período de amostragem  $T_v$ , o qual deve ser definido previamente para cada aplicação, de tal forma que seu valor depende, sobretudo, do passo de simulação<sup>10</sup> (*time step*). Assim, dado um passo de simulação  $t_s$  e um período de amostragem  $T_v$ , tal que  $T_v \leq t_s$ , os valores de orientação  $\alpha_k$  e de velocidade  $v_k$  utilizados para prever as amostras de localização do nodo  $d$  no instante de tempo  $k$  são, respectivamente, definidos pelas Equações 29 e 30:

$$\alpha_k = \alpha_k^{\prime d} \quad (29)$$

$$v_k = \frac{v_{k-1}^{\prime d} + v_{(k-1)+T_v}^{\prime d} + v_{(k-1)+2T_v}^{\prime d} + \dots + v_{(k-1)+(t_s-T_v)}^{\prime d}}{S}, T_v \leq t_s \quad (30)$$

onde  $\alpha_k^{\prime d}$  e  $v_k^{\prime d}$  indicam, respectivamente, a orientação e a velocidade instantâneas do nodo  $d$  medidas no instante de tempo  $k$  e  $S$  indica o número de medições de velocidade realizadas. Conforme pode ser observado, é possível que algumas mudanças de velocidade ocorridas dentro de um intervalo de simulação não sejam verificadas (o mesmo ocorre para mudanças de orientação). Assim, quanto menor o período de amostragem  $T_v$ , mais fidedigno é o valor calculado para  $v_k$ .

Não é foco do presente trabalho a análise acerca dos sensores que podem ser utilizados para a obtenção dos valores de orientação  $\alpha_k^{\prime}$  e de velocidade  $v_k^{\prime}$ . Apesar disso, é válido destacar que dispositivos como acelerômetros são capazes de medir ambas as grandezas, orientação e velocidade, conforme descrito em (FREESCALE SEMICONDUCTOR, 2007).

Caso não seja possível validar amostras geradas a partir da Equação 23, o esquema proposto opera da mesma forma que o algoritmo OTMCL, ou seja, as amostras  $s_k^{\prime i}$  são selecionadas diretamente de uma *anchor box* definida pelas Equações 18 e 19 e filtradas de acordo com as condições especificadas pela Equação 13. Na ausência de nodos âncoras ou de amostras válidas obtidas a partir da *anchor box*, de maneira semelhante ao algoritmo OTMCL, o método proposto utiliza os valores de orientação  $\alpha_k$ , conforme Equação 29, e de velocidade  $v_k$ , conforme Equação 30, para estimar a atual localização do nodo desconhecido. Tal procedimento, por sua vez, contribui para um aumento na precisão das estimativas de localização em comparação aos métodos MCL, MCB e OTMCL.

---

<sup>10</sup> Intervalo de tempo entre cada instante de execução do algoritmo.

Assim, definido o número  $N$  de amostras necessárias para estimar a localização dos nodos desconhecidos, as etapas de predição e de filtragem do esquema proposto são realizadas conforme pseudocódigo comentado no Algoritmo 3.

---

Algoritmo 3 - Pseudocódigo comentado das etapas de predição e de filtragem do esquema de localização proposto.

---

**1.** O número de tentativas de obtenção de uma amostra válida para cada uma das amostras do conjunto  $S_{k-1}$  deve ser definido previamente.

**while** (size( $S_k$ ) <  $N$ ) **do**

**1.1. Predição:**  $S'_k = \{s_k^i \mid s_k^i \text{ é selecionada da distribuição de probabilidade } p(\mathbf{x}_k \mid Z^{k-1}) \text{ (Equação 23) para todo } 1 \leq i \leq N, \text{ onde a área de predição } A \text{ é definida através da Equação 28}\}$

**1.2. Filtragem:**  $S_k = \{s_k^i \mid s_k^i \in S'_k \text{ e } filter(s_k^i) > 0\}$  (Equação 13)

**end**

**2. Estimativa das localizações dos nodos desconhecidos**

**2.1.** Caso  $S_k \neq \emptyset$

$$x_k^{est} = \frac{\sum_{i=1}^{size(S_k)} x_k^i}{size(S_k)}, \text{ onde } x_k^i \text{ é a coordenada } x \text{ da amostra } s_k^i;$$

$$y_k^{est} = \frac{\sum_{i=1}^{size(S_k)} y_k^i}{size(S_k)}, \text{ onde } y_k^i \text{ é a coordenada } y \text{ da amostra } s_k^i;$$

$$l_k^{est} \leftarrow (x_k^{est}, y_k^{est});$$

**2.2.** Caso  $S_k = \emptyset$

**2.2.1. Caso haja nodos âncoras vizinhos de 1 ou 2 hops – Anchor Box**

$$x_{min} = \max_{j=1}^n (x_j - r) \quad x_{max} = \min_{j=1}^n (x_j + r) \quad (18)$$

$$y_{min} = \max_{j=1}^n (y_j - r) \quad y_{max} = \min_{j=1}^n (y_j + r) \quad (19)$$

OBS.: Aqui, o número de tentativas de obtenção de uma amostra válida para cada uma das amostras do conjunto  $S'_k$  é o mesmo definido no passo 1.

**while** (size( $S_k$ ) <  $N$ ) **do**

$$x_k^i \leftarrow \text{rand}[x_{min}, x_{max}];$$


---

---

```

 $y_k^i \leftarrow \text{rand}[y_{\min}, y_{\max}];$ 
 $s_k^i \leftarrow (x_k^i, y_k^i);$ 
if  $\text{filter}(s_k^i) > 0$  (Filtragem através da Equação 13)
     $S_k \leftarrow s_k^i;$ 
end
end
if  $S_k \neq \emptyset$ 
     $x_k^{est} = \frac{\sum_{i=1}^{\text{size}(S_k)} x_k^i}{\text{size}(S_k)};$ 
     $y_k^{est} = \frac{\sum_{i=1}^{\text{size}(S_k)} y_k^i}{\text{size}(S_k)};$ 
     $l_k^{est} \leftarrow (x_k^{est}, y_k^{est});$ 
end

```

### 2.2.2. Caso $S_k$ permaneça vazio ou não haja nodos âncoras vizinhos de 1 ou 2 hops

Utilizam-se os valores de orientação  $\alpha_k$  (Equação 29) e de velocidade  $v_k$  (Equação 30) aplicados sobre a última estimativa de localização ( $l_{k-1}^{est}$ ) do nodo.

$$l_k^{est} = (\alpha_k, v_k) \rightarrow l_{k-1}^{est}$$


---

Por meio das mudanças sugeridas na forma como são realizadas as etapas de predição e de filtragem dos algoritmos baseados no método MCL, objetiva-se:

- o aumento da precisão das estimativas de localização e, conseqüentemente, a redução do erro médio de estimativa;
- a redução do tempo de computação necessário para a obtenção de  $N$  amostras válidas;
- a redução da dependência de nodos âncoras para a realização das estimativas de localização com precisão satisfatória.

### 4.3 RESUMO DO CAPÍTULO

Neste capítulo apresentou-se o esquema de localização proposto, o qual sugere a utilização da técnica de localização *range-free* baseada na contagem de saltos para auxiliar o processo de localização durante a etapa inicial do algoritmo MCL para RSSFs móveis, bem como a inclusão de medições de velocidade e de orientação ao modelo de mobilidade utilizado durante a etapa de predição do algoritmo.



## 5 FERRAMENTA DE VALIDAÇÃO DOS RESULTADOS

Conforme já mencionado, segundo (OLIVEIRA, 2009), algoritmos de localização podem ser validados, basicamente, por meio de experimentos físicos, modelos analíticos e/ou através do uso de simuladores.

Diante da grande diversidade de aplicações e dos inúmeros parâmetros a serem considerados no projeto e na implementação de RSSFs bem como dos algoritmos sobre elas desenvolvidos, a validação de resultados por meio de simuladores torna-se indispensável, sendo utilizada na grande maioria dos trabalhos científicos na área. Nesse sentido, conforme exposto previamente nesta dissertação, (LEVIS, LEE, *et al.*, 2003) afirma que os simuladores cumprem um importante papel ao possibilitarem avaliar alternativas de projeto, explorar configurações difíceis de serem implementadas na prática e observar interações que também se mostram difíceis de serem capturadas em um sistema real. Entretanto, para o caso específico de aplicações em RSSFs móveis, há determinadas métricas, medidas e parâmetros de entrada para os quais os simuladores frequentemente utilizados na literatura (NS-2, TOSSIM e OM-NeT++) não oferecem suporte e/ou flexibilidade.

Diante disso, Hu e Evans analisam os resultados obtidos com o algoritmo apresentado em (HU e EVANS, 2004) por meio de um simulador desenvolvido em linguagem de programação Java (ORACLE, 1977) especialmente para tal propósito, o qual se encontra disponível em (UNIVERSITY OF VIRGINIA, 2004). Os demais trabalhos (BAGGIO e LANGENDOEN, 2008) (MARTINS, CHEN e SEZAKI, 2009) construídos sobre o método MCL original, em geral, têm seus resultados analisados por meio de versões estendidas da ferramenta desenvolvida por Hu e Evans, adaptando-a conforme as especificidades de cada pesquisa.

No presente trabalho, também se utilizou uma extensão do simulador disponibilizado em (UNIVERSITY OF VIRGINIA, 2004), alterando-se, no entanto, a plataforma de desenvolvimento da ferramenta. Ao invés da plataforma Java, optou-se pela utilização do *software* MatLab (MATHWORKS, 2012), o qual é amplamente empregado nos meios acadêmico e profissional. Assim, o código disponibilizado em (UNIVERSITY OF VIRGINIA, 2004) foi simplesmente transcrito para a linguagem de programação do *software* MatLab, recebendo as alterações necessárias em função da mudança de plataforma. A opção por tal mudança justifica-se, sobretudo, pelas alterações propostas no método MCL, às quais exigem o controle sobre determinados parâmetros relacionados à movimentação dos nodos bem como a implementação da técnica de localização baseada na contagem de saltos. Já a opção pelo *software* Ma-

tlab deve-se, principalmente, à sua simplicidade de utilização, facilidade para geração de gráficos, bem como à forma através da qual esse *software* opera vetores e matrizes, estruturas de dados utilizadas pela grande maioria das linguagens de programação de alto nível conhecidas atualmente.

Assim, o capítulo que segue destina-se à apresentação da ferramenta desenvolvida, detalhando seu funcionamento e apresentando os resultados de simulação obtidos para o algoritmo MCL.

## 5.1 FUNCIONAMENTO DA FERRAMENTA DESENVOLVIDA

O funcionamento da ferramenta de simulação desenvolvida no *software* MatLab divide-se, basicamente, em duas etapas:

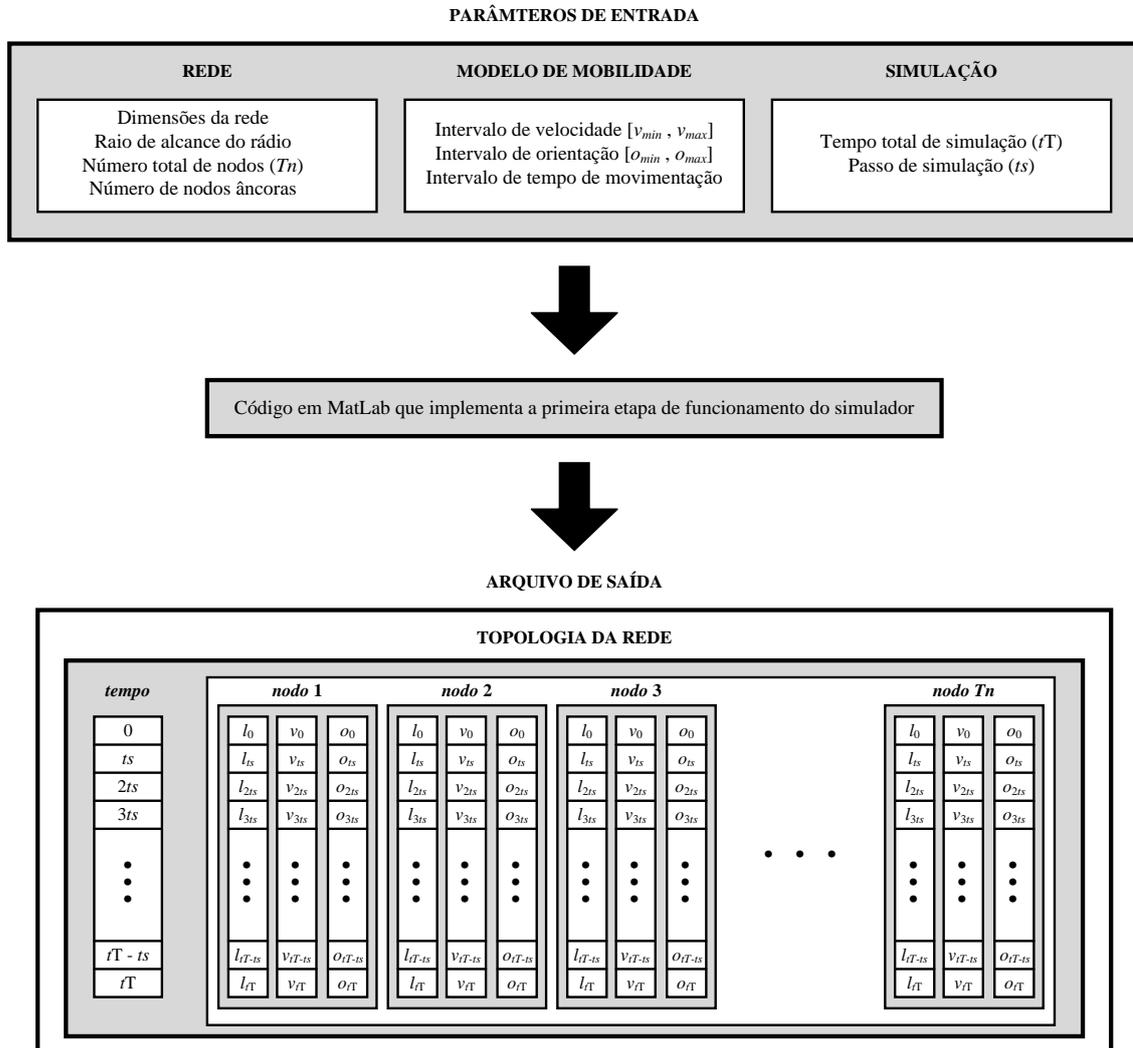
1. construção da topologia da rede para cada um dos instantes de simulação;
2. simulação do algoritmo de localização considerado.

### 5.1.1 Construção da topologia da rede

Durante a primeira etapa de funcionamento, a partir das informações fornecidas pelos parâmetros de entrada listados na Figura 28, o simulador gera a configuração da rede para o tempo total de simulação considerado. Em outras palavras, a saída do simulador fornece a topologia da rede para cada um dos instantes de simulação definidos pelo passo de simulação adotado. Os dados de saída, por sua vez, são disponibilizados no formato da estrutura matricial também mostrada na Figura 28, a qual contém as informações de localização, orientação e velocidade de todos os nodos da rede para cada instante de simulação.

O uso de uma matriz para representar a rede permite que o arquivo de saída da ferramenta desenvolvida possa ser utilizado para a realização de simulações envolvendo RSSFs (móveis e estáticas) com a grande maioria das linguagens de programação de alto nível conhecidas atualmente, tendo em vista a compatibilidade de tais linguagens com a estrutura de dados mencionada. Dessa forma, o código que implementa a primeira etapa de funcionamento do simulador mostra-se uma ferramenta bastante didática, versátil (dadas as informações que disponibiliza acerca da rede) e de fácil utilização (necessita apenas dos parâmetros de entrada).

Figura 28 - Diagrama da primeira etapa de funcionamento do simulador desenvolvido.



Fonte: Autor.

Tendo em vista que o principal objetivo do simulador desenvolvido é a validação dos resultados obtidos com o esquema de localização construído sobre o método MCL original, as configurações adotadas para essa ferramenta de simulação foram implementadas com base no simulador desenvolvido por Hu e Evans. Dessa forma, assim como a ferramenta disponibilizada em (UNIVERSITY OF VIRGINIA, 2004), o simulador aqui apresentado não implementa qualquer modelo de propagação do sinal de rádio. Em outras palavras, assume-se que as mensagens transmitidas por um nodo qualquer da rede são recebidas instantaneamente dentro de sua vizinhança. Diante disso, tal simulador mostra-se incapaz de realizar uma análise fidedigna acerca da quantidade de mensagens transmitidas e/ou do consumo de energia da rede.

Do mesmo modo, o modelo de mobilidade implementado no simulador desenvolvido é o mesmo considerado por Hu e Evans nas análises apresentadas em (HU e EVANS, 2004).

Trata-se de um modelo de mobilidade aleatória *Random Waypoint* modificado, de tal forma que o tempo de pausa associado ao movimento dos nodos móveis é nulo.

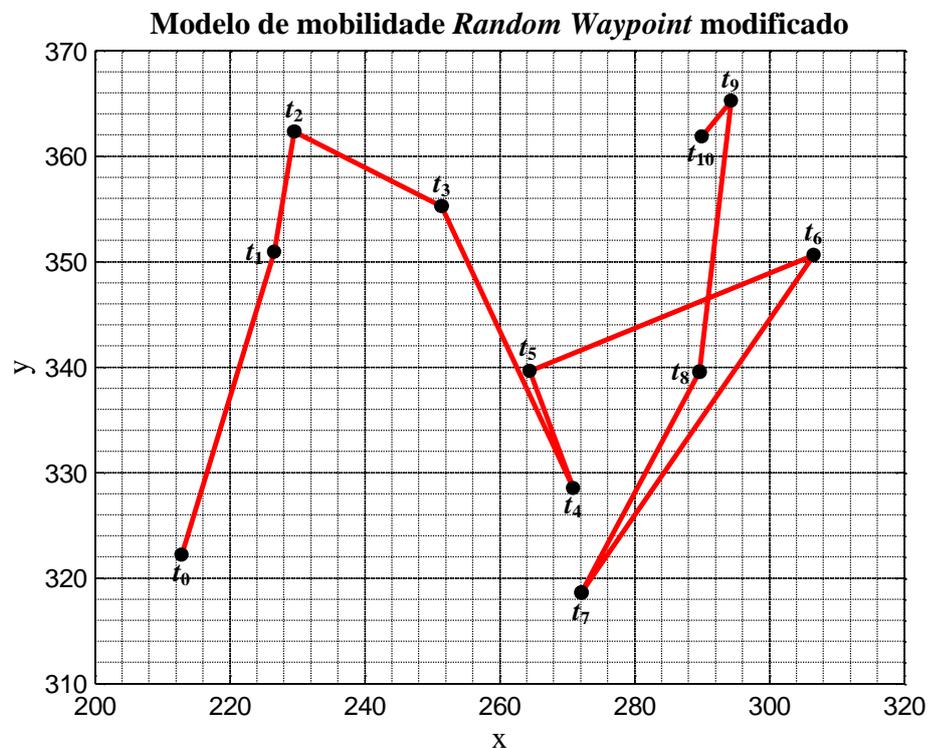
Tradicionalmente, o *Random Waypoint Mobility Model* (JOHNSON e MALTZ, 1996) inclui pausas entre mudanças de orientação e de velocidade associadas ao movimento de um nodo móvel. Assim, o nodo permanece estático em uma determinada localização durante um período de tempo (tempo de pausa). Ao término desse período, o nodo volta a se movimentar, escolhendo um destino (orientação) aleatório dentro da área considerada bem como uma velocidade aleatória dentro do intervalo  $[v_{min}, v_{max}]$ , onde  $v_{min}$  corresponde à velocidade mínima e  $v_{max}$  corresponde à velocidade máxima que pode ser assumida pelo nodo. Entretanto, conforme apontado em (YOON, LIU e NOBLE, 2003), o *Random Waypoint Mobility Model* caracteriza-se por apresentar um decaimento da velocidade média dos nodos móveis, o que acaba gerando uma base inconsistente para a realização das simulações do algoritmo proposto por Hu e Evans bem como para os demais métodos de localização construídos sobre o método MCL. Nesses casos, portanto, opta-se pela adoção de um modelo de mobilidade do tipo *Random Waypoint* modificado, a fim de manter constante a velocidade média de movimentação dos nodos. Para isso, ao invés de escolher uma velocidade para cada destino, os nodos da rede variam aleatoriamente suas velocidades durante cada intervalo de movimentação. Além disso, o tempo de pausa associado ao modelo de mobilidade é igualado à zero, de tal forma que a velocidade média é exatamente  $v_{max}/2$  quando o intervalo de valores de velocidade é definido entre  $[0, v_{max}]$ .

A Figura 29, obtida através do simulador desenvolvido, ilustra a trajetória percorrida por um nodo móvel cujo modelo de mobilidade aleatória do tipo *Random Waypoint* considera as modificações mencionadas (trata-se, neste caso, do modelo de mobilidade implementado no simulador). Os intervalos de tempo  $t_k - t_{k-1}$  representam os intervalos de movimentação do nodo, não estando relacionados aos intervalos de simulação (definidos pelo passo de simulação). Na Tabela 2, são listados os parâmetros de entrada assumidos na simulação.

Tabela 2 - Parâmetros utilizados para simular o modelo de mobilidade implementado.

PARÂMETROS DE SIMULAÇÃO: MODELO DE MOBILIDADE <i>RANDOM WAYPOINT</i> MODIFICADO	
Área de simulação	500 x 500 unidades de distância
Número total de nodos	10
Velocidade máxima dos nodos	50 unidades de distância/passos de simulação
Tempo total de simulação	10 passos de simulação
Modelo de mobilidade	<i>Random Waypoint</i> modificado ( $t_{pausa} = 0$ )

Fonte: Autor.

Figura 29 - Modelo de mobilidade *Random Waypoint* modificado.

Fonte: Autor.

Através da análise da Figura 29, evidentemente, não é possível observar a ausência do tempo de pausa e as alterações de velocidade durante os intervalos de movimentação dos nodos. Essas mudanças no modelo de mobilidade ficam comprovadas a partir dos resultados obtidos com a simulação do algoritmo MCL, os quais são apresentados na próxima seção.

Assim, dados os parâmetros de entrada associados ao modelo de mobilidade, conforme mostrado na Figura 28, o simulador gera, para cada nodo da rede, vetores contendo informações de localização, velocidade e orientação associadas aos instantes de tempo determina-

dos pelos intervalos de movimentação escolhidos aleatoriamente pelos nodos durante o tempo de simulação (intervalos de tempo  $t_k - t_{k-1}$  mostrados na Figura 29). Por fim, com o intuito de obter as informações mencionadas para cada um dos instantes de tempo definidos pelo passo de simulação adotado, realiza-se uma interpolação<sup>11</sup>  $\mathbf{y}_i = \mathbf{interp}(\mathbf{x}, \mathbf{y}, \mathbf{x}_i)$  onde são obtidos os valores  $\mathbf{y}_i(\mathbf{x}_i)$  a partir dos valores em  $\mathbf{y}(\mathbf{x})$ . Nesse caso,  $\mathbf{x}$  e  $\mathbf{y}$  são vetores de mesmo tamanho, assim como  $\mathbf{x}_i$  e  $\mathbf{y}_i$ . O método de interpolação adotado, na verdade, consiste em um procedimento do tipo *lookup table*<sup>12</sup>, onde a tabela, neste caso, é dada pela matriz  $[\mathbf{x}, \mathbf{y}]$ , e a função de interpolação, com base no passo de simulação e no tempo total de simulação, procura os elementos de  $\mathbf{x}_i$  em  $\mathbf{x}$ , e, baseada nas localizações de tais elementos, retorna valores  $\mathbf{y}_i$  interpolados entre os elementos de  $\mathbf{y}$ .

Para a ferramenta de simulação desenvolvida, o vetor  $\mathbf{y}$  representa os vetores contendo as informações de localização, orientação e velocidade dos nodos para cada um dos instantes de tempo contidos no vetor  $\mathbf{x}$ , os quais são gerados com base nos intervalos de movimentação escolhidos pelos nodos. O vetor  $\mathbf{y}_i$ , por sua vez, representa os vetores contendo as mesmas informações de localização, orientação e velocidade dos nodos para cada um dos instantes de tempo contidos no vetor  $\mathbf{x}_i$ , os quais são definidos com base no passo de simulação considerado. Dessa forma, assim como ocorre em (HU e EVANS, 2004), é possível trabalhar com o tempo discretizado.

Tabela 3 - Parâmetros de simulação utilizados para ilustrar a primeira etapa de funcionamento da ferramenta desenvolvida.

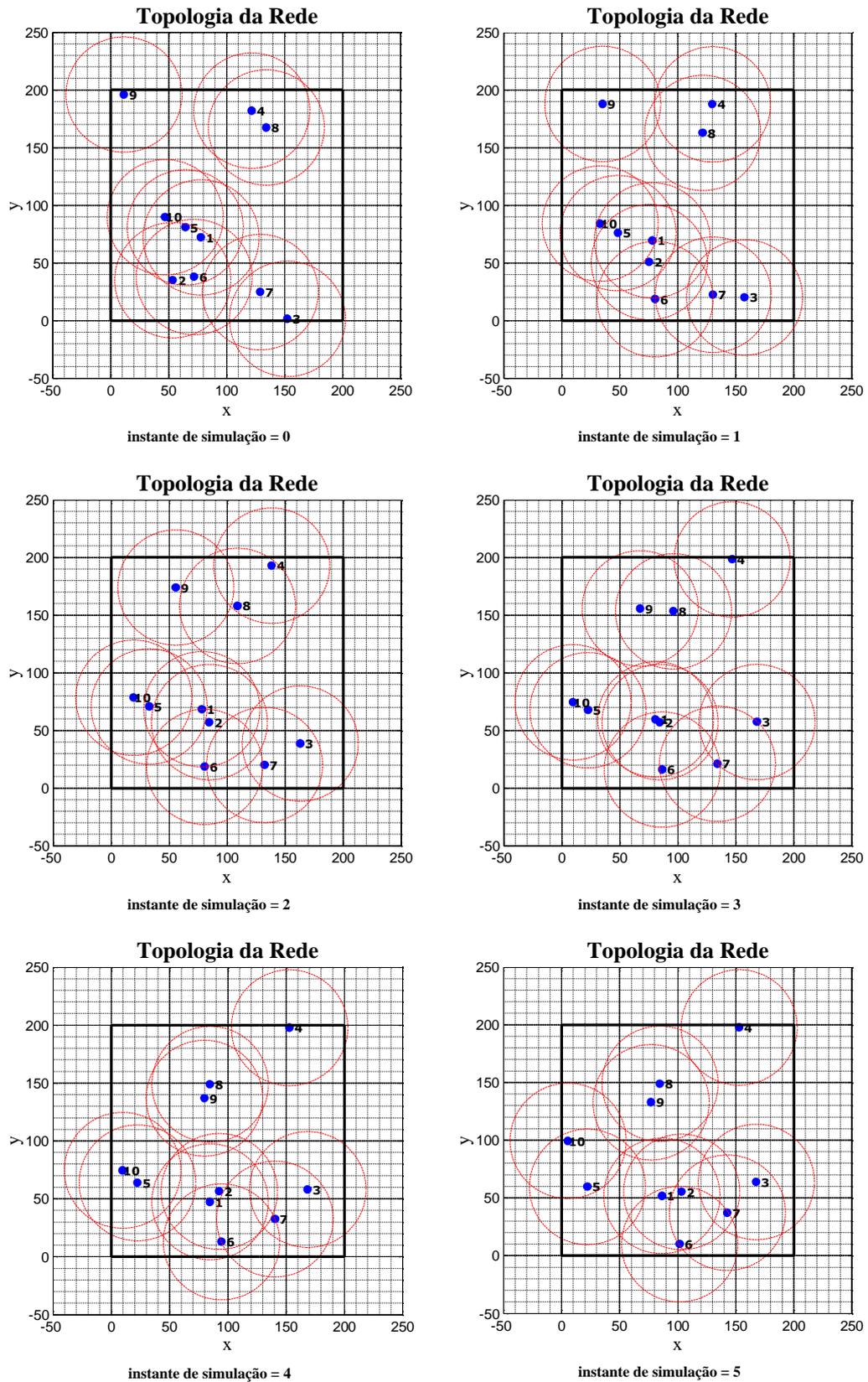
PARÂMETROS DE SIMULAÇÃO: PRIMEIRA ETAPA DE FUNCIONAMENTO	
Área de simulação	200 x 200 unidades de distância
Número total de nodos	10
Velocidade máxima dos nodos	50 unidades de distância/passos de simulação
Alcance de transmissão dos nodos	50 unidades de distância
Tempo total de simulação	5 passos de simulação
Modelo de mobilidade	<i>Random Waypoint</i> modificado ( $t_{pausa} = 0$ )

Fonte: Autor.

<sup>11</sup> Método que permite construir um novo conjunto de dados a partir de um conjunto discreto de pontos previamente conhecido.

<sup>12</sup> Uma *lookup table* (LUT) consiste em uma estrutura de dados (tabela) cujos valores são consultados a fim de se obter determinados resultados de maneira mais rápida e simples, sem a necessidade de cálculos.

Figura 30 - Gráficos gerados pelo simulador desenvolvido mostrando a topologia da rede para cada um dos instantes de simulação definidos pelo passo de simulação.



Fonte: Autor.

A Figura 30 traz os gráficos gerados pelo simulador após a execução da primeira etapa de funcionamento, mostrando a topologia da rede para cada um dos instantes de simulação definidos pelo passo de simulação. Os parâmetros de entrada associados à simulação ilustrada na Figura 30 são exibidos na Tabela 3.

### 5.1.2 Simulação do algoritmo de localização

Obtida a topologia da rede para cada um dos instantes de simulação, é possível implementar o algoritmo de localização desejado sobre as informações fornecidas pelo arquivo de saída mostrado na Figura 28. Obviamente, o algoritmo a ser simulado na ferramenta desenvolvida, caso opere sobre RSSFs móveis, deve considerar o tempo discretizado bem como desconsiderar as perdas de propagação do sinal de rádio. Conforme mencionado na seção anterior, tal algoritmo pode ser implementado em qualquer linguagem de programação compatível com a estrutura de dados do arquivo disponibilizado ao final da primeira etapa de funcionamento do simulador. No presente trabalho, os algoritmos são implementados através do próprio *software* Matlab.

O objetivo da ferramenta de simulação apresentada neste capítulo, conforme já mencionado, consiste em analisar os resultados obtidos com o esquema de localização proposto, o qual é construído sobre o método MCL original. Assim, todas as características da ferramenta desenvolvida são herdadas do simulador utilizado por Hu e Evans na análise do algoritmo apresentado pelos autores em (HU e EVANS, 2004). Por isso, a presente seção traz uma análise comparativa entre os resultados obtidos por meio da utilização das duas ferramentas mencionadas. O algoritmo simulado, nesse caso, é o próprio algoritmo MCL. Da mesma forma como procedido para o modelo de mobilidade considerado, a implementação do algoritmo MCL na ferramenta desenvolvida dá-se pela transcrição do código disponibilizado em (UNIVERSITY OF VIRGINIA, 2004) para a plataforma MatLab.

A análise comparativa entre os resultados de simulação obtidos é feita por meio de gráficos gerados por ambos os simuladores, uma vez que os resultados alcançados com o algoritmo MCL são apresentados em (HU e EVANS, 2004) através de tal recurso.

As principais medidas de análise de desempenho a serem observadas nesta dissertação dizem respeito à convergência da precisão das estimativas de localização ao longo do tempo bem como à variação do erro de estimativa em função da densidade de nodos âncoras da rede. Dessa forma, através da simulação do algoritmo MCL, compara-se:

- o erro de localização das estimativas em função do tempo de simulação;

- o erro de localização das estimativas em função da densidade de nodos âncoras da rede.

### 5.1.2.1 Simulação do algoritmo de localização de Monte Carlo para RSSFs móveis

Tendo em vista o objetivo comparativo dos resultados mostrados a seguir, as simulações do algoritmo de localização de Monte Carlo através do simulador desenvolvido foram feitas com base nos mesmos dados utilizados nas simulações apresentadas em (HU e EVANS, 2004).

Um aspecto importante a ser ressaltado na análise dos resultados diz respeito ao fato de o método de Monte Carlo trabalhar com números aleatórios. Tal aleatoriedade se faz presente não apenas na geração das amostras de localização do algoritmo durante a etapa de predição, mas também na disposição e na movimentação de nodos âncoras e desconhecidos dentro da área da rede. Dessa forma, dificilmente duas simulações apresentarão o mesmo resultado, mesmo sendo feitas utilizando uma única ferramenta de simulação. Ainda em relação a isso, da mesma forma como ocorre em (HU e EVANS, 2004), os valores mostrados nos gráficos gerados por meio do simulador desenvolvido correspondem à média dos resultados obtidos em dez simulações distintas.

A Figura 31 mostra o gráfico que relaciona o erro médio de estimativa do algoritmo (normalizado em relação ao raio de alcance  $r$  dos nodos) com o tempo de simulação. Nesse caso, o gráfico da Figura 31(a) corresponde aos resultados obtidos por meio da ferramenta de simulação desenvolvida neste trabalho, enquanto o gráfico da Figura 31(b) mostra os resultados apresentados em (HU e EVANS, 2004), obtidos por meio do simulador proposto por Hu e Evans em (UNIVERSITY OF VIRGINIA, 2004).

O erro médio de estimativa  $\delta$ , por sua vez, é dado pela Equação 31 (MARTINS, CHEN e SEZAKI, 2009):

$$\delta = \frac{\sum_{n=1}^N \sqrt{(\dot{x}_n - \hat{x}_n)^2 + (\dot{y}_n - \hat{y}_n)^2}}{N \times r} \quad (31)$$

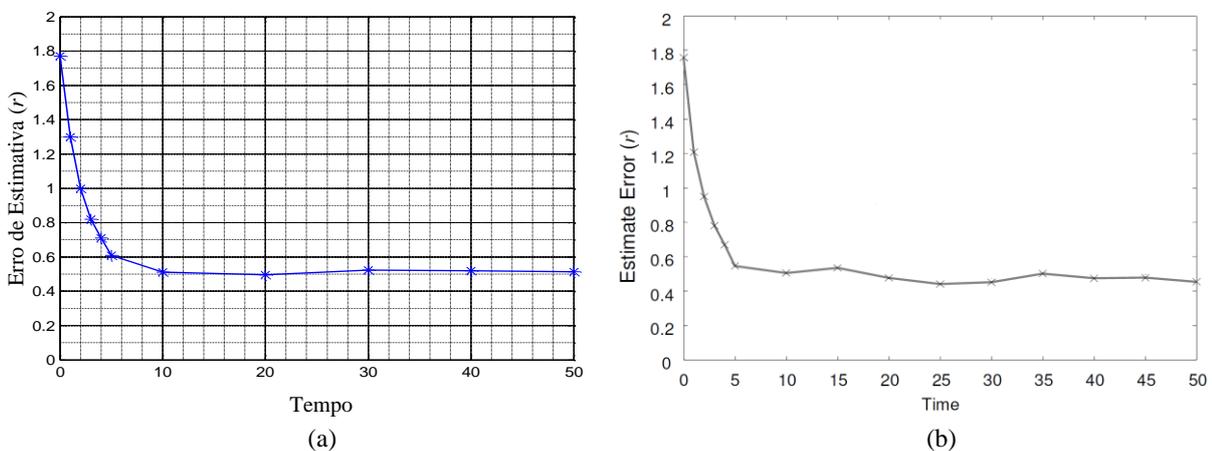
onde  $\dot{x}_n$  e  $\dot{y}_n$  representam as coordenadas estimadas do nodo  $n$ ,  $\hat{x}_n$  e  $\hat{y}_n$  representam as coordenadas reais do nodo  $n$ , e  $N$  é o número total de nodos desconhecidos da rede.

Tabela 4 - Parâmetros de simulação utilizados na análise comparativa entre a ferramenta implementada no software MatLab e aquela disponibilizada em (UNIVERSITY OF VIRGINIA, 2004).

PARÂMETROS DE SIMULAÇÃO: COMPARAÇÃO ENTRE FERRAMENTAS DE SIMULAÇÃO	
Área de simulação	500 x 500 unidades de distância
Número total de nodos	320
Densidade de nodos	10
Número de nodos âncoras	32
Densidade de nodos âncoras	1
Velocidade máxima dos nodos	50 unidades de distância/passos de simulação
Alcance de transmissão dos nodos	50 unidades de distância
Número de amostras	50
Tempo total de simulação	50 passos de simulação
Modelo de mobilidade	<i>Random Waypoint</i> modificado ( $t_{pausa} = 0$ )

Fonte: Autor.

Figura 31 - Comparação entre simuladores por meio da relação entre o erro médio de estimativa do algoritmo MCL e o tempo de simulação. (a) Gráfico obtido através da ferramenta de simulação desenvolvida no presente trabalho; (b) Gráfico obtido através do simulador disponível em (UNIVERSITY OF VIRGINIA, 2004).



Fonte: Autor.

Os dados de simulação utilizados para a obtenção dos resultados mostrados na Figura 31(a) (mesmos dados utilizados para os resultados mostrados na Figura 31(b), conforme (HU e EVANS, 2004)) são listados na Tabela 4.

A Figura 32, por sua vez, traz a variação do erro médio de estimativa do algoritmo MCL em função da densidade de nodos âncoras da rede, de tal forma que os valores mostra-

dos no gráfico da Figura 32(a) são referentes ao simulador desenvolvido neste trabalho, enquanto o gráfico da Figura 32(b) é obtido por meio da ferramenta disponibilizada em (UNIVERSITY OF VIRGINIA, 2004).

O erro médio de estimativa  $\bar{\delta}$ , neste caso, considera o tempo total de simulação e é obtido através da Equação 32:

$$\bar{\delta} = \frac{\sum_{k=1}^{tT} \delta_k}{tT} \quad (32)$$

onde  $\delta_k$  representa o erro médio de estimativa verificado no instante de tempo  $k$  por meio da Equação 31, e  $tT$  é o tempo total de simulação.

A densidade de nodos âncoras ( $a_d$ ) é definida como sendo a quantidade média de nodos âncoras existente em 1 *hop* de transmissão da rede. O número aproximado de *hops* de transmissão dentro da área da rede pode ser obtido dividindo-se a área total da rede pela área de 1 *hop* de transmissão (área da circunferência de raio igual ao raio de alcance dos nodos da rede). Assim, a densidade de nodos âncoras da rede é dada pela Equação 33:

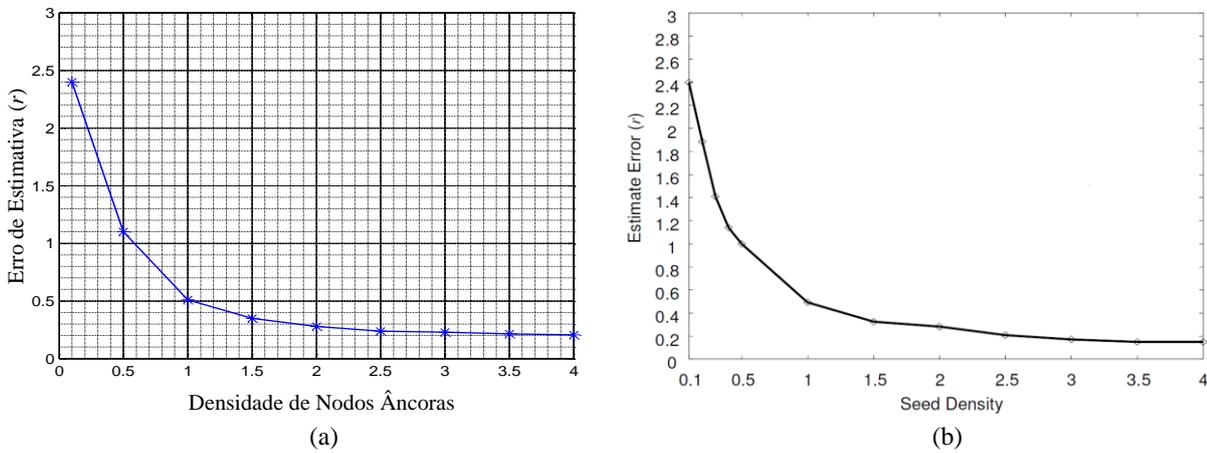
$$a_d = \frac{\pi r^2 n_a}{Area\ da\ rede} \quad (33)$$

onde  $n_a$  é o número total de nodos âncoras da rede.

Os dados de simulação utilizados para a obtenção do gráfico mostrado na Figura 32(a) são os mesmos exibidos na Tabela 4, com exceção da quantidade (densidade) de nodos âncoras, já que esse é o parâmetro variável no gráfico da Figura 32(a).

A partir da análise dos valores plotados nos gráficos das Figuras 31 e 32, observa-se a equivalência entre os resultados obtidos por meio do simulador implementado neste trabalho e aqueles obtidos através da ferramenta disponibilizada em (UNIVERSITY OF VIRGINIA, 2004), provando a eficiência da ferramenta de simulação implementada na análise do erro médio de estimativa em função do tempo e da densidade de nodos âncoras em simulações envolvendo o método de localização de Monte Carlo para RSSFs móveis.

Figura 32 - Comparação entre simuladores por meio da relação entre o erro de estimativa do algoritmo MCL e a densidade de nodos âncoras da rede. (a) Gráfico obtido através da ferramenta de simulação desenvolvida no presente trabalho; (b) Gráfico obtido através do simulador disponível em (UNIVERSITY OF VIRGINIA, 2004).



Fonte: Autor.

## 5.2 RESUMO DO CAPÍTULO

O presente capítulo destinou-se à apresentação da ferramenta de simulação implementada neste trabalho, detalhando suas etapas de funcionamento e expondo os resultados das simulações realizadas com o algoritmo de localização de Monte Carlo para RSSFs móveis. Através da análise comparativa entre os resultados obtidos por meio do simulador desenvolvido neste trabalho e aqueles obtidos através da ferramenta disponibilizada em (UNIVERSITY OF VIRGINIA, 2004), prova-se a eficiência da ferramenta de simulação implementada a fim de validar os resultados alcançados com o esquema de localização proposto no Capítulo 4.

## 6 ANÁLISE DOS RESULTADOS

Neste capítulo do trabalho serão apresentados os resultados obtidos com o esquema de localização proposto no Capítulo 4. Segundo (OLIVEIRA, 2009), a métrica mais importante para se avaliar qualquer algoritmo de localização é o erro de localização, o qual é obtido através da Equação 34 mostrada abaixo. Em geral, os erros de estimativa dos algoritmos de localização para RSSFs são normalizados em relação ao alcance de transmissão  $r$  dos nodos da rede, conforme Equação 31 já apresentada.

$$EL = \sqrt{(X_{estimado} - X_{real})^2 + (Y_{estimado} - Y_{real})^2} \quad (34)$$

Uma medida conveniente para a métrica mencionada é a média do erro sobre todos os nodos desconhecidos da rede, dada pela Equação 31.

Outra medida bastante útil para se avaliar o desempenho de algoritmos de localização para RSSFs diz respeito à densidade de nodos âncoras necessária para a obtenção de estimativas de localização com um erro satisfatório. A densidade de nodos âncoras está intimamente ligada ao custo de implementação e ao consumo de energia dos nodos da rede, uma vez que o aumento do número de nodos de referência significa o aumento do número de dispositivos GPS.

Nas seções que seguem, analisa-se o desempenho do esquema de localização proposto em comparação com alguns dos métodos de localização derivados do método MCL encontrados na literatura. Com base no exposto durante os parágrafos iniciais deste capítulo, tal análise comparativa dá-se, basicamente, pela avaliação das seguintes medidas:

- erro de localização (precisão das estimativas) em função do tempo de simulação;
- erro de localização em função da densidade de nodos âncoras da rede.

Além disso, o esquema proposto também é analisado individualmente através de um estudo acerca do impacto de variações no parâmetro  $\varepsilon$  (referente ao erro de medição da velocidade instantânea do nodo) e no número de amostras  $N$  sobre o erro de estimativa do método.

Conforme apresentado no Capítulo 4, o esquema de localização proposto sugere alterações tanto na etapa de inicialização quanto na etapa de predição do método MCL, de tal forma que cada uma das mudanças sugeridas visa melhorar o desempenho do método de localização em pontos específicos. Assim, com base em tais objetivos, a análise de desempenho é feita, inicialmente, apenas para a etapa de inicialização, e, posteriormente, considerando todas as etapas de funcionamento do esquema proposto.

## 6.1 RESULTADOS OBTIDOS COM A ETAPA DE INICIALIZAÇÃO

A etapa de inicialização corresponde ao método utilizado pelo correspondente algoritmo de localização para estimar as coordenadas dos nodos desconhecidos no instante de tempo  $k = 0$ . Dessa forma, considerando a etapa de inicialização de maneira isolada, pode-se dizer que a mesma opera sobre uma rede estática cuja topologia é definida de acordo com as coordenadas  $x$  e  $y$  de cada um dos nodos da rede no instante de tempo  $k = 0$ . Dessa forma, o desempenho dessa etapa de funcionamento dos algoritmos comparados não é função do tempo.

Tabela 5 - Parâmetros de simulação utilizados para avaliar a etapa de inicialização do ELP e dos métodos MCL, MCB e OTMCL.

PARÂMETROS DE SIMULAÇÃO: ETAPA DE INICIALIZAÇÃO	
Área de simulação	500 x 500 unidades de distância
Número total de nodos	320
Densidade de nodos	10
Número de nodos âncoras	Variável de 3 a 128
Densidade de nodos âncoras	Variável de 0.1 a 4.0
Alcance de transmissão dos nodos	50 unidades de distância
Número de amostras	50
Modelo de mobilidade	Rede estática

Fonte: Autor.

Com base nos resultados pretendidos (Seção 4.1) com as alterações propostas à etapa de inicialização do método MCL, as medidas a serem analisadas na avaliação do desempenho dessa etapa do esquema proposto são:

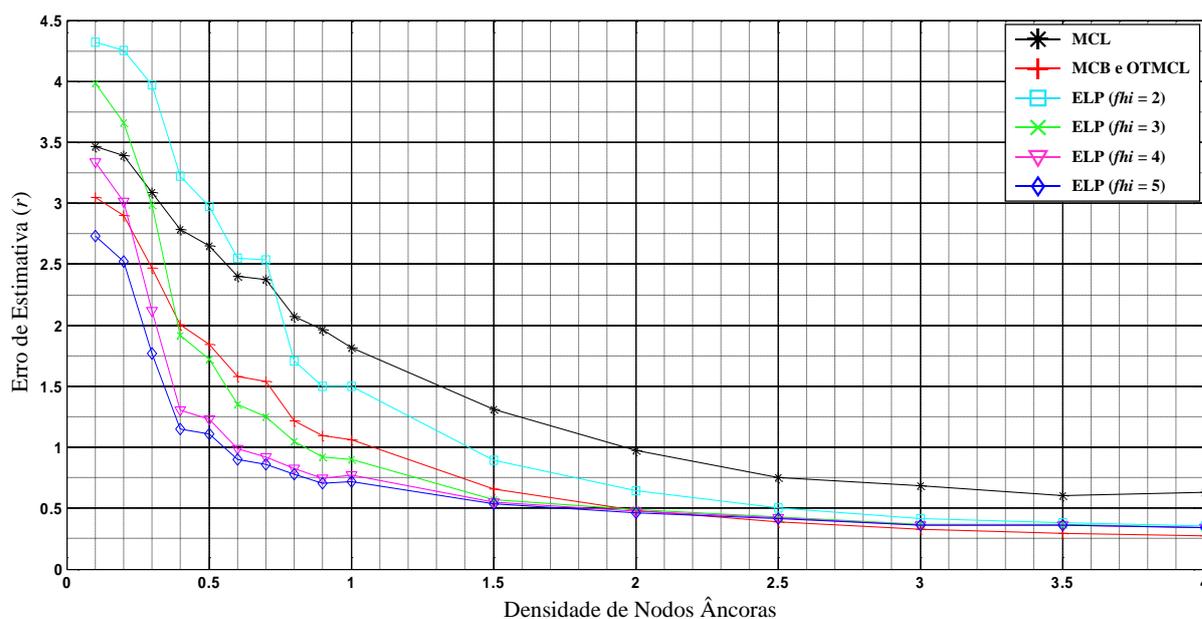
- erro médio de localização, dado pela Equação 31, em função da densidade de nodos âncoras da rede;
- quantidade de nodos desconhecidos sem estimativa de localização.

Os resultados obtidos com o esquema proposto são comparados com aqueles obtidos com os algoritmos MCL, MCB e OTMCL, tendo em vista que todos esses algoritmos são construídos sobre o método de localização de Monte Carlo para RSSFs.

Os parâmetros de simulação utilizados para a obtenção dos resultados exibidos ao longo desta seção são listados na Tabela 5.

O gráfico mostrado na Figura 33 traz os valores do erro médio de estimativa (Equação 31), medido como um múltiplo do alcance de transmissão  $r$ , obtido para cada um dos algoritmos comparados em função da densidade de nodos âncoras da rede, considerando os quatro esquemas de localização operando sobre o mesmo cenário. Os valores mostrados no gráfico foram obtidos através do cálculo da média dos valores alcançados em dez simulações. Em outras palavras, foram gerados dez cenários distintos e, para cada um desses cenários, foram simulados todos os esquemas de localização comparados. Para as simulações envolvendo o esquema de localização proposto (ELP), consideraram-se ainda diferentes valores para o parâmetro  $fhi$ .

Figura 33 - Impacto da densidade de nodos âncoras sobre o erro médio de estimativa durante a etapa de inicialização do esquema de localização proposto (ELP) e dos algoritmos MCL, MCB e OTMCL.



Fonte: Autor.

Conforme mencionado durante a formulação do esquema de localização proposto, o valor atribuído ao parâmetro  $fhi$  depende da densidade total de nodos da rede bem como da densidade de nodos de referência. A relação existente entre o valor de  $fhi$  e a densidade de nodos âncoras da rede é inversa, uma vez que para uma quantidade maior de nodos âncoras é possível diminuir o alcance ( $fhi$ ) das mensagens propagadas por tais nodos sem que ocorra a redução do número de nodos desconhecidos cobertos por nodos âncoras. Tal relação pode ser observada através do comportamento decrescente das curvas do esquema de localização pro-

posto (ELP) mostradas no gráfico da Figura 33. Dessa forma, para valores baixos de densidade de nodos âncoras, a regulação correta do valor de  $fhi$  garante que a etapa de inicialização do esquema de localização proposto apresente um erro médio de estimativa inferior aos demais métodos simulados. Essa, de fato, é a grande vantagem das alterações sugeridas para a etapa de inicialização do esquema proposto. Deve-se levar em consideração, entretanto, o aumento do consumo de energia da rede associado ao aumento do valor de  $fhi$ , uma vez que este representa o aumento do número de mensagens transmitidas dentro da rede. Tal análise de consumo, no entanto, não faz parte do escopo do presente trabalho.

Já para valores elevados de densidade de nodos âncoras, o valor de  $fhi$  influencia na precisão das estimativas do esquema proposto, dado que um aumento de  $fhi$  significa um aumento no número de informações de referência que integram o processo de estimativa de localização dos nodos desconhecidos. Entretanto, conforme exposto em (NICULESCU e NATH, 2001), para faixas de valores elevados de densidade de nodos âncoras, os algoritmos de localização baseados na técnica de contagem de saltos não apresentam uma variação significativa no erro de estimativa, visto que a cobertura da rede por parte dos âncoras e o número de informações utilizadas no processo de estimativa mantêm-se praticamente constantes. Tal fato também pode ser observado diante da análise das curvas do esquema de localização proposto (ELP) mostradas na Figura 33, onde se verifica que os erros de estimativa de localização das 4 curvas apresentadas tendem para o mesmo valor à medida que a densidade de nodos âncoras aumenta.

A Tabela 6 mostra os valores correspondentes à redução do erro médio de estimativa durante a etapa de inicialização do ELP em comparação aos métodos MCL, MCB e OTMCL, considerando uma faixa de valores de densidade de nodos âncoras que varia de 0.1 à 1.5. Os dados exibidos na tabela foram obtidos através das mesmas simulações cujos resultados são mostrados na Figura 33. Todos os valores mostrados estão associados ao menor valor de  $fhi$  que garante ao ELP um erro de estimativa inferior ao verificado para o algoritmo em comparação. Além disso, os valores são expressos como porcentagens dos respectivos valores de erro dos algoritmos MCL, MCB e OTMCL.

Tabela 6 - Redução do erro médio de estimativa durante a etapa de inicialização do ELP em comparação aos métodos MCL, MCB e OTMCL.

DENSIDADE DE NODOS ÂNCORAS						
0.1	0.2	0.3	0.4	0.5	1.0	1.5
<b>MCL</b>						
3.74 <sub>fhi = 4</sub>	11.15 <sub>fhi = 4</sub>	3.31 <sub>fhi = 3</sub>	31.19 <sub>fhi = 3</sub>	34.99 <sub>fhi = 3</sub>	17.41 <sub>fhi = 2</sub>	31.73 <sub>fhi = 2</sub>
<b>MCB e OTMCL</b>						
10.48 <sub>fhi = 5</sub>	13.06 <sub>fhi = 5</sub>	14.34 <sub>fhi = 4</sub>	4.23 <sub>fhi = 3</sub>	6.53 <sub>fhi = 3</sub>	15.05 <sub>fhi = 3</sub>	13.22 <sub>fhi = 3</sub>

Fonte: Autor.

Ainda para fins comparativos, a Tabela 7 traz o percentual de nodos desconhecidos não cobertos por nodos de referência durante a etapa de inicialização dos algoritmos comparados, considerando diferentes valores de densidade de nodos âncoras. Os valores exibidos na tabela foram obtidos através das mesmas simulações cujos resultados são mostrados na Figura 33, e, da mesma forma, são dados pela média aritmética simples dos valores alcançados em 10 simulações distintas. A medida apresentada na Tabela 7, por si só, não constitui uma métrica para avaliação de desempenho de algoritmos de localização em RSSFs. Os valores mostrados, entretanto, auxiliam na interpretação dos resultados mostrados no gráfico da Figura 33.

Tabela 7 - Percentual de nodos desconhecidos sem estimativa de localização durante a etapa de inicialização do ELP e dos métodos MCL, MCB e OTMCL.

ALGORITMO	DENSIDADE DE NODOS ÂNCORAS									
	0.1	0.2	0.3	0.4	0.5	1.0	1.5	2.0	2.5	3.0
<b>MCL</b>	88	84	76	67	64	41	25	17	13	12
<b>MCB e OTMCL</b>	70	62	50	35	31	12	4	2	1	1
<b>ELP (<math>f_{hi} = 2</math>)</b>	70	62	50	35	31	12	4	2	1	1
<b>ELP (<math>f_{hi} = 3</math>)</b>	50	40	28	14	12	3	1	1	0	0
<b>ELP (<math>f_{hi} = 4</math>)</b>	30	24	14	4	4	1	0	0	0	0
<b>ELP (<math>f_{hi} = 5</math>)</b>	16	12	7	1	1	0	0	0	0	0

Fonte: Autor.

A etapa de inicialização dos métodos de localização MCB e OTMCL é a única, dentre os métodos comparados, a realizar a filtragem das amostras geradas a partir das chamadas *anchor boxes*. Por esse motivo, embora os valores da Tabela 7 associados aos métodos MCB

e OTMCL para as densidades 0.1 e 0.2 sejam muito superiores aos respectivos valores do ELP com  $fhi = 4$ , o erro médio de estimativa dos primeiros algoritmos mencionados ainda é inferior ao erro verificado para o ELP.

Vale ressaltar que, para todos os esquemas de localização apresentados, os nodos desconhecidos que não recebem mensagens de nodos âncoras durante a etapa de inicialização têm suas posições estimadas no ponto central da área da rede. Portanto, aumentar a quantidade de nodos desconhecidos cobertos por nodos de referência contribui para a redução do erro médio de estimativa do ELP em comparação aos demais métodos (considerando o erro durante os instantes iniciais de simulação), mesmo que as estimativas de localização para alguns nodos desconhecidos não sejam tão precisas.

## 6.2 RESULTADOS OBTIDOS COM O ESQUEMA DE LOCALIZAÇÃO PROPOSTO

Os resultados apresentados nesta seção consideram todas as etapas de funcionamento do ELP, ou seja, as etapas de inicialização, predição e filtragem. Com base nos resultados pretendidos com as alterações sugeridas (Seção 4.2), as medidas a serem analisadas na avaliação do desempenho do esquema proposto são:

- o impacto do parâmetro  $\varepsilon$  (associado ao erro de medição de velocidade) sobre a precisão das estimativas;
- o erro de estimativa em função do tempo de simulação (precisão das estimativas);
- o erro médio de estimativa, dado pela Equação 32, em função da densidade de nodos âncoras da rede;
- o impacto da quantidade  $N$  de amostras sobre a precisão das estimativas.

As avaliações acerca do impacto do parâmetro  $\varepsilon$  e da quantidade de amostras sobre o erro de estimativa são feitas por meio de uma análise individual do ELP, ou seja, sem a realização de comparações com outros métodos de localização. Para as demais medidas verificadas, da mesma forma como procedido na análise dos resultados obtidos com a etapa de inicialização, os resultados alcançados com o esquema proposto são comparados com aqueles obtidos com os algoritmos MCL, MCB e OTMCL.

### 6.2.1 O impacto do parâmetro $\varepsilon$ sobre o erro de estimativa

Aqui, o ELP é analisado isoladamente a fim de se verificar o impacto que o parâmetro  $\varepsilon$  (associado ao erro de medição de velocidade) exerce sobre a precisão das estimativas do algoritmo e, conseqüentemente, sobre o erro médio de estimativa do mesmo.

Os parâmetros de simulação utilizados para realizar tal análise são listados na Tabela 8 abaixo.

Tabela 8 - Parâmetros de simulação utilizados para avaliar o impacto do parâmetro  $\varepsilon$  sobre o erro médio de estimativa do ELP.

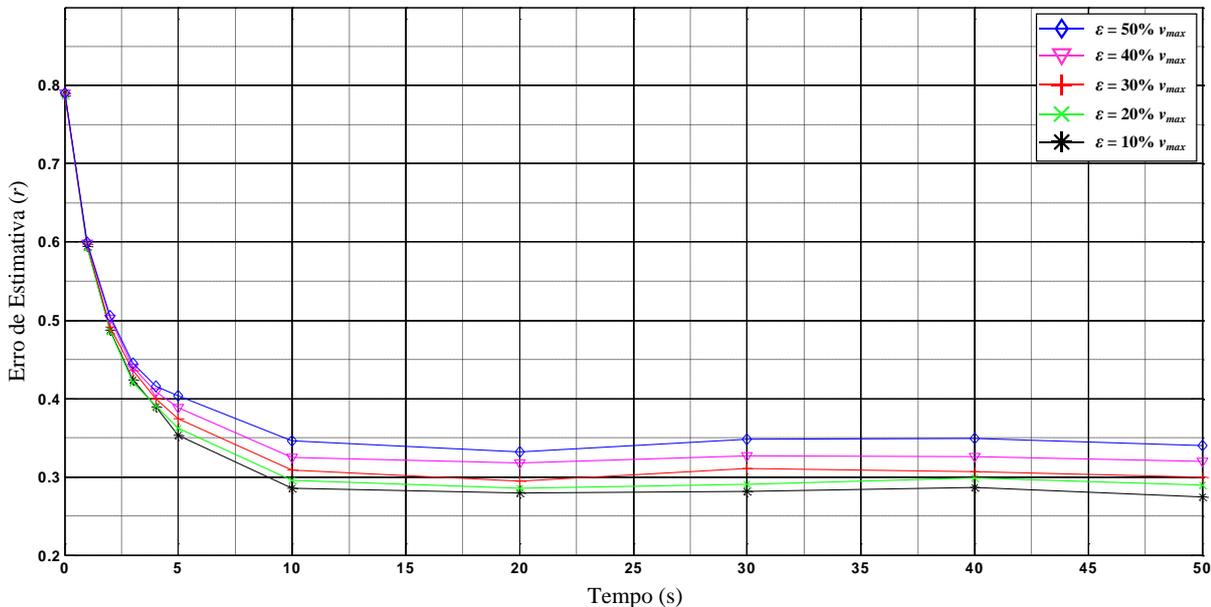
PARÂMETROS DE SIMULAÇÃO: IMPACTO DE $\varepsilon$ SOBRE O ERRO DE ESTIMATIVA	
Área de simulação	500 x 500 unidades de distância
Número total de nodos	320
Densidade de nodos	10
Número de nodos âncoras	32
Densidade de nodos âncoras	1
Velocidade máxima dos nodos	50 unidades de distância/passos de simulação
Alcance de transmissão dos nodos	50 unidades de distância
Número de amostras	50
Modelo de mobilidade	<i>Random Waypoint</i> modificado ( $t_{pausa} = 0$ )
Tempo total de simulação	50 passos de simulação
$f_{hi}$	5
$\beta$	45°
$\varepsilon$	Variável de 10% de $v_{max}$ a 50% de $v_{max}$
Medições de velocidade no intervalo de simulação	2

Fonte: Autor.

O gráfico da Figura 34 traz os valores de erro de estimativa do ELP, medido como um múltiplo do alcance de transmissão  $r$ , em função do tempo, considerando 5 valores distintos para o parâmetro  $\varepsilon$ . Os valores de  $\varepsilon$ , por sua vez, são dados como porcentagens da velocidade máxima  $v_{max}$  dos nodos da rede. O impacto do parâmetro  $\beta$  (associado ao erro de medição de orientação) não é analisado neste trabalho, visto que tal estudo é apresentado em (MARTINS, CHEN e SEZAKI, 2009). Assim, as curvas apresentadas no gráfico da Figura 34 são obtidas para um valor fixo de  $\beta = 45^\circ$ . Da mesma forma como ocorre para as simulações feitas na se-

ção anterior, os valores mostrados no gráfico abaixo foram obtidos através do cálculo da média dos valores alcançados em dez simulações distintas.

Figura 34 - Precisão das estimativas do ELP em função do tempo para diferentes valores de  $\varepsilon$ .



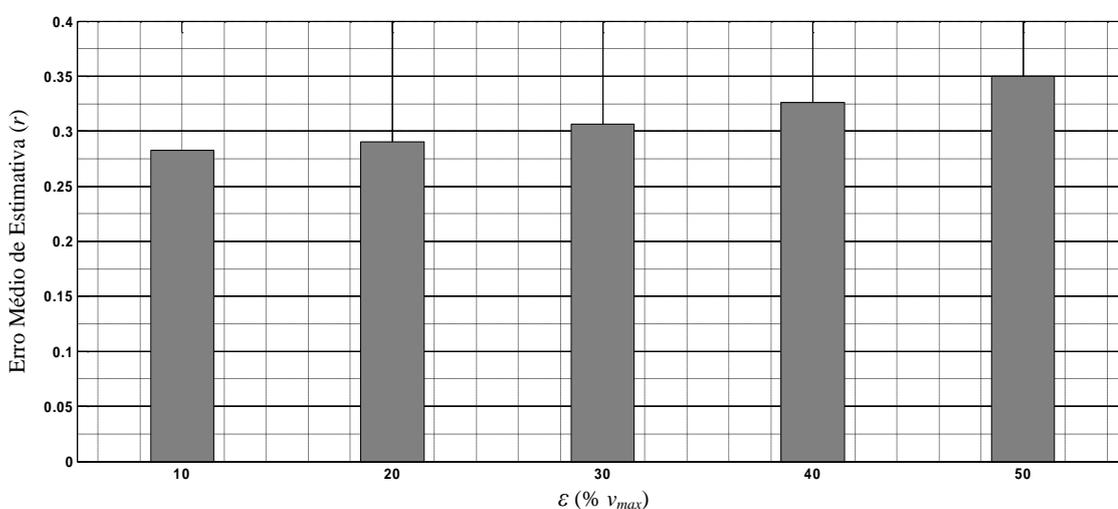
Fonte: Autor.

A precisão das estimativas dos esquemas de localização baseados no método de Monte Carlo está diretamente associada ao tamanho da área de predição considerada pelo algoritmo, de tal forma que áreas menores abrangem amostras mais precisas acerca da localização do dispositivo. Isso explica a redução do erro de estimativa do ELP à medida que o valor de  $\varepsilon$  diminui, conforme evidencia a análise das curvas mostradas no gráfico da Figura 34. Essa mesma relação também pode ser observada no estudo acerca da variação do erro de estimativa em função do parâmetro  $\beta$  do algoritmo OTMCL.

Tanto a Figura 26 quanto a Equação 28 podem ser utilizadas para visualizar a influência que variações do parâmetro  $\varepsilon$  exercem sobre o tamanho da área de predição do ELP, variação esta que irá refletir no comportamento do erro de estimativa. Comparativamente, variações do parâmetro  $\beta$  produzem maiores variações no erro de estimativa do algoritmo em relação a variações proporcionais do parâmetro  $\varepsilon$ , conforme exposto em (MARTINS, CHEN e SEZAKI, 2009). Isso ocorre porque a porção de área de predição que se altera através de uma variação em  $\beta$  é maior do que a porção alterada por uma variação proporcional em  $\varepsilon$ .

De forma a tornar tal análise mais precisa, a Figura 35 traz um gráfico em barras mostrando o valor do erro médio de estimativa do ELP para alguns valores de  $\varepsilon$  (exibidos como porcentagens da velocidade máxima  $v_{max}$  dos nodos). Os dados mostrados no gráfico abaixo são obtidos a partir dos valores exibidos no gráfico da Figura 34, considerando apenas a fase estável das curvas, ou seja, o ponto de performance nominal (para as curvas exibidas, consideram-se os valores de erro obtidos a partir dos 10 segundos de simulação).

Figura 35 - Erro médio de estimativa do ELP para diferentes valores de  $\varepsilon$ .



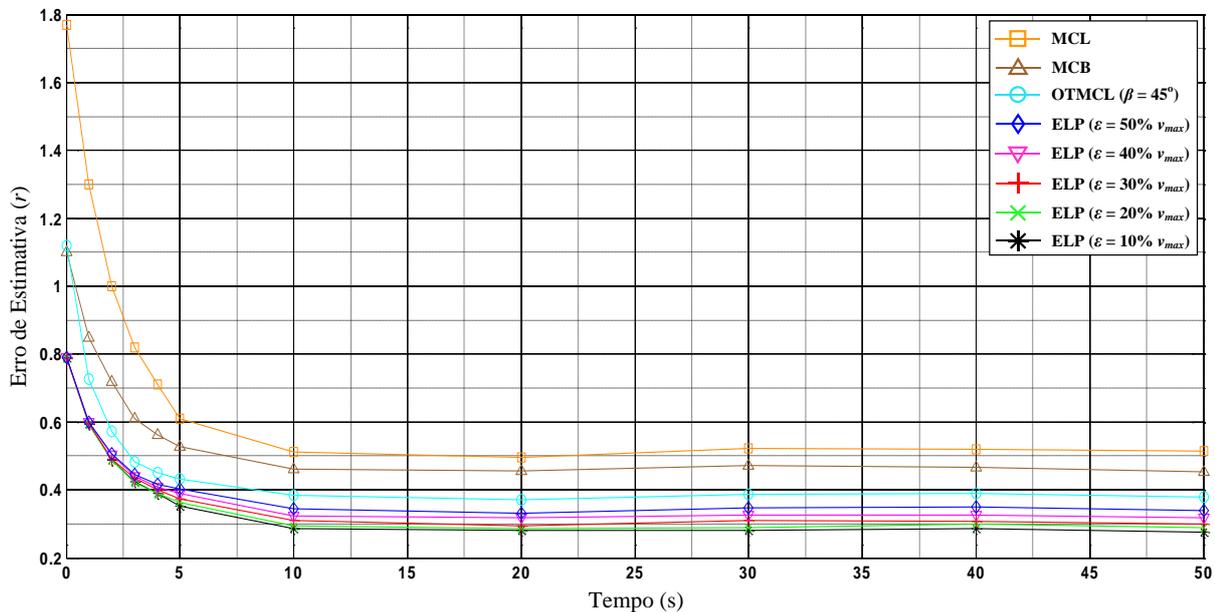
Fonte: Autor.

### 6.2.2 Precisão das estimativas de localização em função do tempo

A segunda análise a ser feita sobre o ELP diz respeito à convergência da precisão das estimativas de localização ao longo do tempo. Para isso, compara-se novamente o ELP com os algoritmos MCL, MCB e OTMCL. Os parâmetros de simulação adotados para tal análise são aqueles mostrados na Tabela 8.

O gráfico da Figura 36, portanto, mostra o comportamento do erro de estimativa, medido como um múltiplo do alcance de transmissão  $r$ , em função do tempo para os quatro esquemas de localização comparados. Da mesma forma como procedido para as demais simulações apresentadas neste trabalho, os valores que compõem as curvas da Figura 36 resultam da média aritmética simples dos valores alcançados em 10 simulações distintas.

Figura 36 - Precisão das estimativas de localização em função do tempo de simulação para o ELP e para os métodos de localização MCL, MCB e OTMCL.



Fonte: Autor.

A queda inicial acentuada verificada em todas as curvas da Figura 36 é característica dos esquemas de localização baseados no método de Monte Carlo e ocorre devido à incorporação das informações de localização dos nodos âncoras ao processo sequencial desse método. Obviamente, a amplitude dessa queda verificada nos instantes iniciais de simulação depende do método de estimativa de localização adotado pelo algoritmo durante a etapa de inicialização, conforme análise feita na primeira seção do presente capítulo.

Após essa fase inicial, as curvas estabilizam-se, evidenciando o equilíbrio entre as atualizações associadas à distribuição de localização posterior e as incertezas introduzidas pela mobilidade dos nodos. Da mesma forma como ocorre para a etapa de inicialização, o valor do erro em torno do qual a curva permanece estável depende da precisão das estimativas do método de localização considerado, conforme evidenciam os valores mostrados no gráfico da Figura 36.

Dentre os três métodos de localização utilizados na comparação com o ELP, aquele que apresenta o melhor desempenho em termos de precisão nas estimativas é o método OTMCL, dada a inclusão da informação de orientação na determinação da área de predição desse algoritmo. Através da análise das curvas mostradas na Figura 36, entretanto, é possível observar que o erro associado à precisão das estimativas do ELP ao longo de todo o tempo de simulação mostra-se inferior aos erros verificados para todos os demais métodos. Seguindo a

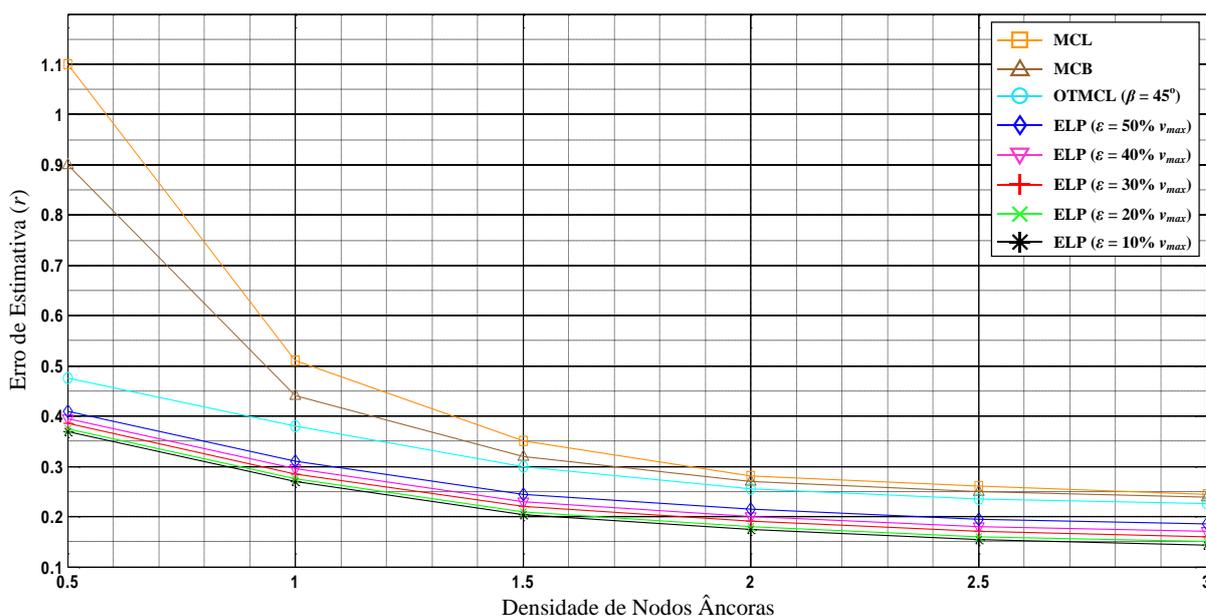
mesma lógica do método OTMCL, essa redução no erro de estimativa ocorre, evidentemente, pela utilização de informações acerca da velocidade dos nodos, o que acaba reduzindo a área de predição sobre a qual atua a função de amostragem do ELP. À medida que o valor do parâmetro  $\varepsilon$  aumenta, o erro de estimativa do ELP no ponto de performance nominal tende a ser o mesmo verificado para o algoritmo OTMCL de mesmo valor  $\beta$  ( $45^\circ$ ), dado que a faixa de valores de velocidade que definem a área de predição do ELP tende para o intervalo  $[0, v_{max}]$ .

Apesar de o erro de estimativa associado ao ELP durante os instantes iniciais de simulação ser inferior ao verificado para os demais métodos, o tempo de convergência observado para todas as curvas é o mesmo. Isso ocorre pelo fato de todos os esquemas simulados valem-se apenas das informações de nodos âncoras vizinhos de 1 e 2 hops, ou seja, a incorporação de novas observações ao processo sequencial de estimativa de localização se dá na mesma proporção para todos os algoritmos.

### 6.2.3 O impacto da densidade de nodos âncoras sobre o erro de estimativa

Nesta seção, avalia-se o impacto da densidade de nodos âncoras da rede sobre o erro médio de estimativa do ELP.

Figura 37 - Impacto da densidade de nodos âncoras sobre o erro de estimativa do esquema de localização proposto (ELP) e dos algoritmos MCL, MCB e OTMCL.



Fonte: Autor.

Os parâmetros de simulação adotados são os mesmos exibidos na Tabela 8, com exceção da densidade (quantidade) de nodos âncoras, uma vez que esse é o parâmetro variável para a análise em questão.

O gráfico da Figura 37, portanto, traz os valores do erro de estimativa, medido como um múltiplo do alcance de transmissão  $r$ , em função da densidade de nodos âncoras da rede para os quatro esquemas de localização comparados. Novamente, todos os valores plotados resultam da média aritmética simples dos valores alcançados em 10 simulações distintas. Além disso, os valores de erro mostrados correspondem ao erro médio de estimativa do método de localização no ponto de performance nominal.

Durante a etapa de inicialização bem como em qualquer instante de tempo no qual o conjunto de amostras de um nodo desconhecido está vazio, o método MCL original apenas utiliza as informações de nodos âncoras vizinhos de 2 hops se estas estiverem combinadas com as informações de âncoras vizinhos de 1 hop. Tal procedimento acaba penalizando a precisão do processo de estimativa desse algoritmo, principalmente em topologias com baixa densidade de nodos âncoras. Nessa mesma situação, tanto o ELP como os métodos MCB e OTMCL permitem que um nodo desconhecido utilize informações provenientes de nodos âncoras situados à 2 hops de distância mesmo que não haja nodos âncoras vizinhos de 1 hop. Nos três últimos esquemas citados, esse procedimento é realizado por meio da construção das *anchor boxes*.

Entretanto, para as situações onde um nodo desconhecido apresenta um conjunto de amostras vazio e, além disso, não possui quaisquer nodos âncoras vizinhos de 1 ou 2 hops, o método MCB procede da mesma forma que o método MCL original, ou seja, posiciona esse nodo desconhecido no ponto central da área da rede. O ELP e o método OTMCL, por sua vez, utilizam as informações de orientação (ELP e OTMCL) e de velocidade (ELP) do nodo desconhecido para estimar sua posição atual. Tal procedimento pode ser feito, sobretudo, porque os dois métodos mencionados possuem informações acerca da orientação do nodo.

Os dois aspectos mencionados acima contribuem significativamente para as diferenças verificadas entre os valores de erro de estimativa dos quatro métodos comparados na Figura 37, principalmente para os valores mais baixos de densidade de nodos âncoras.

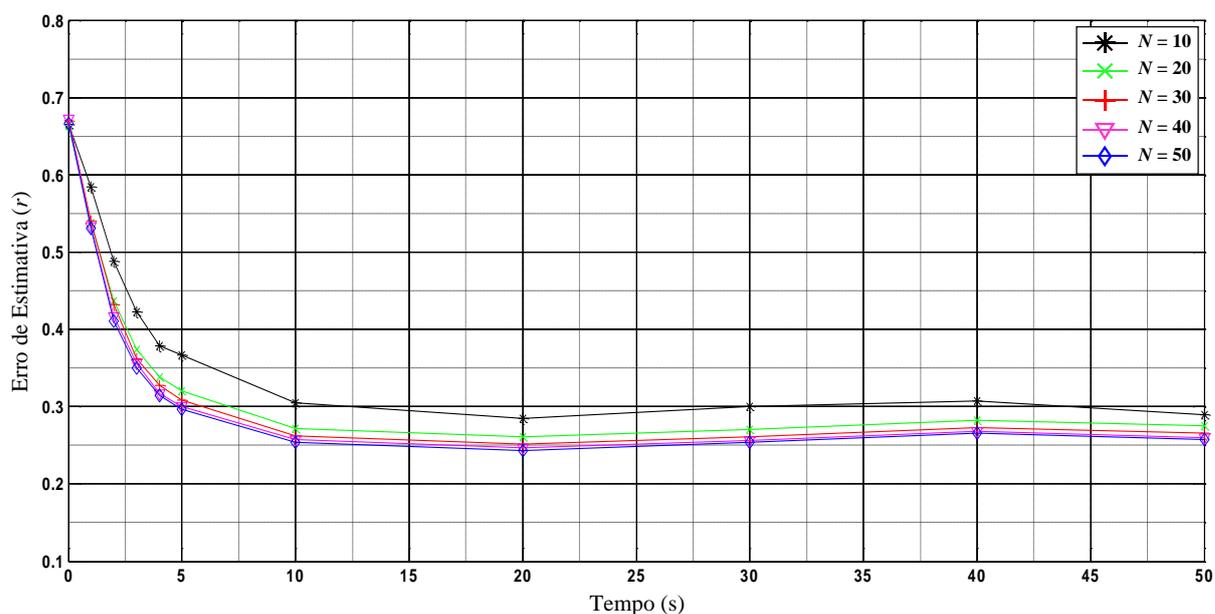
Por fim, os menores valores de erro de estimativa do ELP em comparação ao método OTMCL devem-se à utilização da informação de velocidade dos nodos pelo ELP durante a etapa de predição, uma vez que a etapa de filtragem se dá da mesma forma para os dois métodos. À medida que se aumenta o valor de  $\varepsilon$ , os valores de erro das curvas do ELP no gráfico da Figura 37 tendem para os valores da curva do método OTMCL de mesmo valor  $\beta$ .

### 6.2.4 O impacto da quantidade de amostras sobre o erro de estimativa

Por fim, o ELP é novamente analisado de maneira isolada, objetivando avaliar o impacto de variações na quantidade  $N$  de amostras obtidas durante as etapas de predição e de filtragem sobre o erro de estimativa do método. Os parâmetros de simulação adotados são os mesmos exibidos na Tabela 8, com exceção da quantidade  $N$  de amostras, uma vez que esse é o parâmetro variável para a análise em questão, e do parâmetro  $\varepsilon$ , o qual é fixado em 10% do valor de  $v_{max}$ .

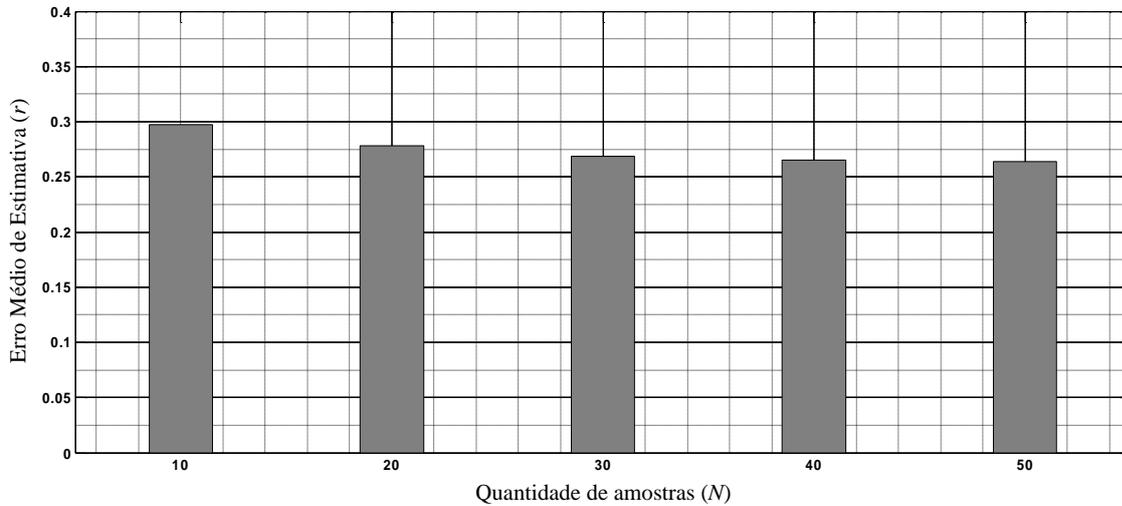
O gráfico da Figura 38 traz os valores do erro de estimativa do ELP, medido como um múltiplo do alcance de transmissão  $r$ , em função do tempo para diferentes valores de  $N$ . Assim como nas simulações anteriores, todos os valores plotados resultam da média aritmética simples dos valores alcançados em 10 simulações distintas.

Figura 38 - Precisão das estimativas do ELP em função do tempo para diferentes valores de  $N$ .



Fonte: Autor.

De forma a tornar a análise mais precisa, a Figura 39 traz um gráfico em barras mostrando o valor do erro médio de estimativa do ELP para alguns valores de  $N$ . Os dados mostrados no gráfico abaixo são obtidos a partir dos valores exibidos no gráfico da Figura 38, considerando apenas a fase estável das curvas, ou seja, o ponto de performance nominal (para as curvas exibidas, consideram-se os valores de erro obtidos a partir dos 10 segundos de simulação).

Figura 39 - Erro médio de estimativa do ELP para diferentes valores de  $N$ .

Fonte: Autor.

Tanto as curvas da Figura 38 quanto o gráfico em barras da Figura 39 mostram a pequena variação do erro de estimativa do ELP em função da quantidade  $N$  de amostras obtidas durante as etapas de predição e de filtragem desse método. A diferença entre o erro obtido com  $N = 10$  e aquele obtido com  $N = 50$  é de apenas 0.034 (valor medido como um múltiplo do alcance de transmissão  $r$ ). Essa pequena variação justifica-se pelo fato de a área de predição utilizada pelo ELP ser bastante reduzida. Dessa forma, mesmo que sejam consideradas poucas amostras, as mesmas apresentam uma precisão considerável, dadas as informações de orientação e de velocidade dos nodos utilizadas durante o processo de amostragem.

A possibilidade de se utilizar uma pequena quantidade de amostras é bastante importante do ponto de vista do custo computacional e, conseqüentemente, do consumo de energia dos nodos. Em outras palavras, uma quantidade inferior de amostras requer um tempo de processamento também inferior.

### 6.3 RESUMO DO CAPÍTULO

No presente capítulo foram apresentados e analisados os resultados obtidos com o esquema de localização proposto. Para tanto, o desempenho do esquema proposto foi comparado com o desempenho dos algoritmos MCL, MCB e OTMCL no que diz respeito ao erro de estimativa em função do tempo de simulação e da densidade de nodos âncoras da rede. Além disso, o esquema proposto também foi analisado individualmente através de um estudo acerca

do impacto de variações no parâmetro  $\varepsilon$  e no número de amostras  $N$  sobre o erro de estimativa.



## 7 CONCLUSÃO

No presente trabalho propôs-se um esquema *range-free* de localização de nodos em RSSFs móveis construído sobre o método de localização de Monte Carlo apresentado em (HU e EVANS, 2004), tendo como objetivo principal a redução do erro das estimativas de localização, especialmente em cenários com baixa densidade de nodos âncoras, visto que o erro de localização é considerado a métrica mais importante para se avaliar um algoritmo de localização (OLIVEIRA, 2009). Para isso, foram realizadas alterações nas etapas de inicialização, através da implementação da técnica de contagem de saltos (NICULESCU e NATH, 2001) (HSIEH e WANG, 2006), e de predição do método de Monte Carlo original, por meio da incorporação de medições de velocidade e de orientação dos nodos. Com base na simulação de cenários com diferentes densidades de nodos âncoras e com nodos movendo-se de acordo com uma versão modificada do modelo de mobilidade *random waypoint* (JOHNSON e MALTZ, 1996), o esquema de localização proposto apresentou um erro de estimativa inferior àquele verificado para alguns dos principais métodos de localização derivados do método de Monte Carlo para RSSFs.

Além disso, desenvolveu-se uma extensão da ferramenta de simulação disponibilizada em (UNIVERSITY OF VIRGINIA, 2004), a qual foi implementada por Hu e Evans (HU e EVANS, 2004) em linguagem de programação Java (ORACLE, 1977). No presente trabalho, entretanto, optou-se pelo uso da plataforma MatLab (MATHWORKS, 2012), tendo em vista a simplicidade de utilização dessa ferramenta, a facilidade para geração de gráficos, e a forma por meio da qual esse *software* opera vetores e matrizes, estruturas de dados utilizadas pela grande maioria das linguagens de programação de alto nível conhecidas atualmente, permitindo portar os arquivos da ferramenta desenvolvida para diferentes plataformas de programação. Através de análises comparativas, verificou-se a eficiência do simulador desenvolvido para realizar a análise dos resultados obtidos com algoritmos de localização para RSSFs móveis derivados do método de Monte Carlo original. Dessa forma, a plataforma de simulação implementada foi utilizada para realizar a validação dos resultados obtidos com o esquema de localização proposto.

Como trabalhos futuros, destaca-se a realização de novas análises acerca do desempenho do esquema de localização proposto, verificando o comportamento do erro de estimativa de localização diante de variações na velocidade máxima dos nodos da rede, na densidade de nodos do cenário considerado e no modelo de mobilidade adotado pelos nodos. Quanto à ferramenta de simulação desenvolvida, cita-se como trabalho futuro a implementação de mode-

los de propagação do sinal de rádio, permitindo a realização de simulações mais fidedignas às condições observadas em cenários reais bem como a realização de análises acerca do consumo de energia associado à troca de mensagens dos algoritmos de localização simulados.

## REFERÊNCIAS

21 ideas for the 21st century. **BusinessWeek**, [S.l.], p. 78-167, August 1999.

AHMED, R.; AVARITSIOTIS, J. N. Mobile element based localization and tracking using path-loss model in WSNs. **Science and Information Conference (SAI)**, London, UK, August 2014.

AKYILDIZ, F. et al. Wireless sensor networks: a survey. **Computer Networks**, [S.l.], v. 38, n. 4, p. 393-422, 15 March 2002.

AMMARI, H. M. **The art of wireless sensor networks - Volume 1: Fundamentals**. [S.l.]: Springer, 2014.

BACHIR, A.; DOHLER, M.; WATTEYNE, T. MAC essentials for wireless sensor networks. **IEEE Communications Surveys & Tutorials**, [S.l.], v. 12, n. 2, p. 222-248, April 2010.

BAGGIO, A.; LANGENDOEN, K. Monte Carlo localization for mobile wireless sensor networks. **Ad Hoc Netw.**, [S.l.], v. 6, p. 718-733, 2008.

BAHL, P.; PADMANABHAN, N. RADAR: an in-building RF-based user location and tracking system. **Proceedings of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)**, [S.l.], March 2000.

BHUVANESWARI, T. V.; BALAKUMAR, R.; VAIDEHI, V. Solar energy harvesting for wireless sensor networks. **First International Conference on Computational Intelligence, Communication Systems and Networks (CICSYN)**, [S.l.], p. 57-61, July 2009.

BISWAS, P.; AGHAJAN, H.; YE, Y. Integration of angle of arrival information for multimodal sensor network localization using semidefinite programming. **39th Asilomar Conference on Signals, Systems and Computers**, [S.l.], November 2005.

BULUSU, N.; HEIDEMANN, J.; ESTRIN, D. GPS-less low-cost outdoor localization for very small devices. **IEEE Personal Communications**, [S.l.], v. 7, n. 5, p. 28-34, October 2000.

BURGARD, W. et al. Estimating the absolute position of a mobile robot using position probability grids. **Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI)**, [S.l.], p. 896-901, 1996.

CABRERA-MORA, F.; XIAO, J. Reprocessing technique to signal strength data of wireless sensor network for real-time distance estimation. **IEEE International Conference on Robotics and Automation (ICRA)**, [S.l.], p. 1537-1542, 2008.

CAMP, T.; BOLENG, J.; DAVIES, V. A survey of mobility models for ad hoc network research. **Wireless Communications & Mobile Computing**, [S.l.], v. 2, n. 5, 2002.

CHAO, J. et al. Performance evaluation of DV-hop localization algorithm with mobility models for mobile wireless sensor networks. **9th International Wireless Communications and Mobile Computing Conference (IWCMC)**, [S.l.], July 2013.

CHEN, Y. et al. Realizing mobile node tracking in wireless sensor network based on Kalman filter. **IEEE 3rd International Conference on Communication Software and Networks (ICCSN)**, [S.l.], May 2011.

CHONG, C. Y.; KUMAR, S. Sensor networks: evolution, opportunities, and challenges. **Proceedings of the IEEE**, [S.l.], v. 91, n. 8, p. 1247-1256, 2003.

CROSSBOW TECHNOLOGY INC. **Open Automation**, 2013. Disponível em: <[http://www.openautomation.net/uploads/productos/micaz\\_datasheet.pdf](http://www.openautomation.net/uploads/productos/micaz_datasheet.pdf)>. Acesso em: 14 Julho 2016.

DARDARI, D. et al. Mathematical Evaluation of Environmental Monitoring Estimation Error through Energy-Efficient Wireless Sensor Networks. **IEEE Transactions on Mobile Computing (TMC)**, [S.l.], v. 6, n. 7, p. 790-802, Julho 2007.

DARDARI, D. et al. Mathematical evaluation of environmental monitoring estimation error through energy-efficient wireless sensor networks. **IEEE Transactions on Mobile Computing**, [S.l.], v. 6, n. 7, p. 790-802, July 2007.

DELLAERT, F. et al. Monte Carlo localization for mobile robots. **IEEE International Conference on Robotics and Automation (ICRA)**, [S.l.], May 1999.

DIL, B.; DULMAN, S.; HAVINGA, P. Range-based localization in mobile sensor networks. **Third European Workshop on Wireless Sensor Networks, Lecture Notes in Computer Science**, [S.l.], n. 3868, p. 164-179, 2006.

DOUCET. On sequential simulation-based methods for bayesian filtering. **Technical report CUED/F-INFENG/TR.310 - University of Cambridge**, 1998.

FREESCALE SEMICONDUCTOR. Implementing positioning algorithms using accelerometers. **Application Note**, February 2007.

FRIEDMAN, D.; HEINRICH, H.; DUAN, D. W. A low-power cmos integrated circuit for field-powered radiofrequency identification tags. **43rd IEEE International Solid-State Circuits Conference (ISSCC)**, [S.l.], p. 294-295, 1997.

GARDNER, V. K. V.; AWADELKARIM, O. O. **Microsensors, MEMS and smart devices**. 1. ed. New York, NY, USA: Wiley, 2001.

GETTING, I. Perspective/navigation - the global positioning system. **IEEE Spectrum**, [S.l.], v. 30, n. 12, p. 36-38, 1993.

GORDON, N. J.; SALMOND, D. J.; SMITH, A. F. M. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. **IEEE Proceedings-F**, [S.l.], v. 140, n. 2, p. 107-113, April 1993.

GRACIOLI, G.; NUNES, R. C. Detectores de defeitos para redes wireless ad hoc. **IX Escola Regional de Redes de Computadores**, Passo Fundo, RS, 2006.

GRAEFENSTEIN, J.; BOUZOURAA, M. Robust method for outdoor localization of a mobile robot using received signal strength in low power wireless networks. **Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)**, [S.l.], p. 33-38, 2008.

GREENORBS. **GreenOrbs**, 2011. Disponivel em: <<http://www.greenorbs.org/all/greenorbs.htm>>. Acesso em: 10 Julho 2016.

GUSTAFSSON, F.; GUNNARSSON, F. **Positioning Using Time-Difference Of Arrival Measurements**, 2003. Disponivel em: <[citeseer.ist.psu.edu/548251.html](http://citeseer.ist.psu.edu/548251.html)>.

HANDSCHIN, J. E. Monte Carlo techniques for prediction and filtering of non-linear stochastic processes. **Automatica** 6, [S.l.], p. 555-563, 1970.

HARTL, G.; LI, B. A Bayesian inference approach towards energy efficient data collection in dense sensor networks. **25th IEEE International Conference on Distributed Computing Systems (ICDCS)**, [S.l.], p. 371-380, 2005.

HE, T. et al. Range-free localization schemes for large scale sensor networks. **Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom)**, New York, NY, USA, p. 81-95, 2003.

HONG, X. et al. A group mobility model for ad hoc wireless networks. **Proceedings of the 2nd ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)**, New York, NY, USA, p. 53-60, 1999.

HSIEH, Y.; WANG, K. Efficient localization in mobile wireless sensor networks. **IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing**, [S.l.], June 2006.

HU, L.; EVANS, D. Localization for mobile sensor networks. **10th Annual International Conference on Mobile Computing and Networking (MobiCom)**, Philadelphia, September 2004.

IEEE. **IEEE std. 802.15.4 - 2003 Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low Rate Wireless Personal Area Networks (LRWPANs)**, 2003. Disponivel em: <<http://standards.ieee.org/getieee802/download/802.15.4-2003.pdf>>. Acesso em: 10 Julho 2016.

ISARD, M.; BLAKE, A. Contour tracking by stochastic propagation of conditional density. **European Conference on Computer Vision**, [S.l.], p. 343-356, 1996.

ISARD, M.; BLAKE, A. Condensation - conditional density propagation for visual tracking. **International Journal of Computer Vision**, [S.l.], v. 29, n. 1, p. 5-28, 1998.

JI, W.-W.; LIU, Z. An improvement of dv-hop algorithm in wireless sensor networks. **International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)**, [S.l.], p. 1-4, 2006.

JOHNSON, D. B.; MALTZ, D. A. Dynamic source routing in ad hoc wireless networks. **Mobile Computing**, [S.l.], v. 353, p. 153-181, 1996.

JUANG, P. et al. Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet. **10th International Conference on Architectural Support for Programming Language (ASPLOS)**, San Jose, CA, October 2002.

KANG, D. et al. A simple asynchronous uwb position location algorithm based on single round-trip transmission. **The 8th International Conference Advanced Communication Technology (ICACT)**, [S.l.], v. 3, p. 4, 2006.

LADD, M. A. et al. Robotics-based location sensing using wireless ethernet. **Wireless Networks**, [S.l.], v. 11, n. 1-2, p. 189-204, 2005.

LANGENDOEN, K.; REIJERS, N. Distributed localization in wireless sensor networks: a quantitative comparison. **The International Journal of Computer and Telecommunications Networking**, [S.l.], v. 43, n. 4, p. 499-518, 2003.

LEVIS, P. et al. TOSSIM: accurate and scalable simulation of entire TinyOS applications. **Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys)**, New York, NY, USA, p. 126-137, 2003.

MACK, C. Fifty years of Moore's law. **IEEE Transactions on Semiconductor Manufacturing**, [S.l.], v. 24, n. 2, p. 202-207, 2011.

MARTINS, L. G. et al. A hybrid method to detecting failures in mobile sensor networks using localization algorithms. **11th IEEE International New Circuits and Systems Conference (NEWCAS)**, Paris, FR, p. 1-4, June 2013.

MARTINS, M. H. T.; CHEN, H.; SEZAKI, K. OTMCL: Orientation Tracking-based Monte Carlo Localization for mobile sensor networks. **Sixth Annual International Conference on Networked Sensing Systems (INSS)**, Pittsburgh, PA, June 2009.

MARTINS, T. G. et al. Desenvolvimento de um equipamento para uso na pecuária leiteira capaz de medir o grau de ruminção e atividade em bovinos. **Jornada Nacional de Iniciação Científica – 65ª Reunião Anual da Sociedade Brasileira para o Progresso da Ciência (SBPC)**, Recife, PE, Brasil, Julho 2013.

MATHWORKS. **MATLAB and Simulink for technical computing**, 2012. Disponível em: <<http://www.mathworks.com>>. Acesso em: 10 Julho 2016.

MAYBECK, P. **Stochastic models, estimation and control**. New York, NY, USA: Academic Press, v. 1, 1979.

MONDINELLI, F.; KOVACS-VAJNA, Z. Self-localizing sensor network architectures. **IEEE Transactions on Instrumentation and Measurement**, [S.l.], v. 53, n. 2, p. 277-283, 2004.

MOZUMDAR, M. et al. A framework for modeling, simulation and automatic code generation of sensor network application. **5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)**, [S.l.], p. 515-522, 2008.

MULLER, C. Método de multilateração para algoritmos de localização em redes de sensores sem fio. **Dissertação (Mestrado) - Universidade Federal de Santa Maria**, 2009.

NAGPAL, R.; SHROBE, H.; BACHRACH, J. Organizing a global coordinate system from local information on an ad hoc sensor network. **2th International Workshop on Information Processing in Sensor Networks (IPSN)**, [S.l.], April 2003.

NASIPURI, A.; LI, K. A directionality based location discovery scheme for wireless sensor networks. **First ACM International Workshop on Wireless Sensor Networks and Applications**, Atlanta, GA, September 2002.

NICULESCU, D.; NATH, B. Ad hoc Positioning System (APS). **IEEE Global Telecommunications Conference (GLOBECOM)**, [S.l.], v. 5, p. 2926-2931, November 2001.

NICULESCU, D.; NATH, B. Ad hoc Positioning System (APS) using AoA. **22nd Annual Joint Conference of the IEEE Computer and Communications (INFOCOM)**, [S.l.], March 2003.

NICULESCU, D.; NATH, B. DV based positioning in ad hoc networks. **Kluwer Journal of Telecommunication Systems**, p. 267-280, 2003.

OLIVEIRA, L. L. Algoritmo de localização de nodos para redes de sensores móveis. **Tese (Doutorado) - Universidade Federal de Santa Maria**, 2009.

ORACLE. **JAVA**, 1977. Disponível em: <<http://www.oracle.com/technetwork/java/index.html>>.

PANTAZIS, N.; VERGADOS, D. A survey on power control issues in wireless sensor networks. **IEEE Communications Surveys and Tutorials**, [S.l.], v. 9, n. 4, p. 86-107, 2007.

PARKINSON, B.; GILBERT, S. Navstar: Global positioning system - ten years later. **Proceedings of the IEEE**, [S.l.], v. 71, n. 10, p. 1177-1186, 1983.

PATWARI, N. et al. Locating the nodes: cooperative localization in wireless sensor networks. **IEEE Signal Processing Magazine**, v. 22, n. 4, p. 54-69, July 2005.

PATWARI, N.; III, A. O. H. Using proximity and quantized rss for sensor localization in wireless networks. **IEEE ACM 2nd Workshop on Wireless Sensor Networks and Applications**, 2003.

PENG, C.; SHEN, G.; ZHANG, Y. BeepBeep: a high-accuracy acoustic-based system for ranging and localization using cots devices. **ACM Transactions on Embedded Computing Systems (TECS)**, New York, NY, USA, v. 11, n. 1, p. 4:1-4:29, April 2012.

RABAEY, J. et al. PicoRadio supports ad hoc ultra-low power wireless networking. **Computer**, [S.l.], v. 33, n. 7, p. 42-48, August 2000.

RAPPAPORT, T. **Wireless Communications - principles and practice**. 2. ed. Upper Saddle River, NJ, USA: Prentice Hall PTR., 2001.

RONG, P.; SICHITIU, M. Angle of arrival localization for wireless sensor networks. **3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks (SECON)**, [S.l.], v. 1, p. 374-382, 2006.

ROYER, E. M.; MELLIAR-SMITH, P. M.; MOSER, L. E. An analysis of the optimum node density for ad hoc mobile networks. **Proceedings of the IEEE International Conference on Communications (ICC)**, [S.l.], p. 857-861, June 2001.

SATHYAN, T.; HEDLEY, M. Fast and accurate cooperative tracking in wireless networks. **IEEE Transactions on Mobile Computing**, [S.l.], v. 12, n. 9, September 2013.

SAVARESE, C.; RABAEY, J. M.; LANGENDOEN, K. Robust positioning algorithms for distributed ad-hoc wireless sensor networks. **Proceedings of the General Track of the Annual Conference on USENIX Annual Technical Conference (ATEC)**, Berkeley, CA, USA, p. 317-327, 2002.

SAVARESE, C.; RABAEY, J.; BEUTEL, J. Location in distributed ad-hoc wireless sensor networks. **IEEE International Conference on Acoustics, Speech, and Signal Processing**, [S.l.], v. 4, p. 2037-2040, 2001.

SAVVIDES, A.; HAN, C.; STRIVASTAVA, M. B. Dynamic fine-grained localization in ad hoc networks of sensors. **Proceedings of the 7th annual international conference on Mobile computing and networking (MobiCom)**, New York, NY, USA, p. 166-179, 2001.

SPENCER, S. J. The two-dimensional source location problem for time differences of arrival at minimal element monitoring arrays. **The Journal of the Acoustical Society of America**, [S.l.], v. 121, n. 6, p. 3579-3594, 2007.

STEHLIK, M. Comparison of simulators for wireless sensor networks. **Dissertação (Mestrado) - Masaryk University Faculty of Informatics**, [S.l.], 2011.

STEVENS-NAVARRO, E.; VIVEKANANDAN, V.; WONG, V. W. Dual and mixture Monte Carlo localization algorithms for mobile wireless sensor networks. **IEEE Wireless Communications and Networking Conference (WCNC)**, [S.l.], p. 4024-4028, 2007.

TEXAS INSTRUMENTS. **CC2420**, 2012. Disponível em: <<http://www.ti.com/lit/ds/symlink/cc2420.pdf>>. Acesso em: 12 Julho 2016.

TIAN, S. et al. A selective anchor node localization algorithm for wireless sensor networks. **International Conference on Convergence Information Technology**, [S.l.], p. 358-362, 2007.

UNIVERSITY OF VIRGINIA, 2004. Disponível em: <<http://www.cs.virginia.edu/mcl>>. Acesso em: 10 Julho 2016.

VIVEKANANDAN, V.; WONG, V. Concentric Anchor-Beacons (CAB) localization for wireless sensor networks. **IEEE International Conference on Communications**, [S.l.], v. 9, p. 3972-3977, 2006.

WANG, T. et al. Continuous tracking for mobile targets with mobility nodes in WSNs. **International Conference on Smart Computing (SMARTCOMP)**, [S.l.], November 2014.

WELLENHOFF, B. H.; LICHTENEGGER, H.; COLLINS, J. **Global positioning system - theory and practice**. 4. ed. [S.l.]: Springer Verlag, 1997.

WILLIAMS, C.; YATES, R. Analysis of a micro-electric generator for microsystems. **International Conference on Solid-State Sensors and Actuators**, [S.l.], p. 369-372, 1995.

XIAO, J.; REN, L.; TAN, J. Research of TDOA based self-localization approach in wireless sensor network. **IEEE/RSJ International Conference on Intelligent Robots and Systems**, [S.l.], p. 2035-2040, 2006.

YOON, J.; LIU, M.; NOBLE, B. Sound mobility models. **Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom)**, New York, NY, USA, p. 205-216, 2003.

YUAN, L.; CHEN, W.; XI, Y. A review of control and localization for mobile sensor networks. **The Sixth World Congress on Intelligent Control and Automation**, [S.l.], v. 2, p. 9164-9168, 2006.

ZHANG, H. et al. Mobile node tracking in an ad hoc network using double time difference of data arrival and Kalman filtering. **2nd International Conference on Signal Processing and Communication Systems**, [S.l.], p. 1-4, 15-17, December 2008.

ZHAO, J. et al. Localization of wireless sensor networks in the wild: pursuit of ranging quality. **IEEE/ACM Transactions on Networking**, [S.l.], v. 21, n. 1, p. 311-323, February 2013.

ZHAO, S.; SUN, M.; ZHANG, Z. GASA-hop localization algorithm for wireless sensor networks. **WRI International Conference on Communications and Mobile Computing (CMC)**, [S.l.], v. 2, p. 152-156, January 2009.

ZHENYU, S. A simulation tool for wireless sensor networks. **Dissertação (Mestrado) - Politecnico di Torino**, Torino, Italy, 2009.