

**Ricardo Reghelin**

**Um Algoritmo Descentralizado de Localização  
para Rede de Sensores Sem Fio Usando  
Calibragem Cooperativa e Heurísticas**

**Florianópolis**

**2007**

# **Livros Grátis**

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

**UNIVERSIDADE FEDERAL DE SANTA CATARINA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM**  
**CIÊNCIA DA COMPUTAÇÃO**

**Ricardo Reghelin**

**Um Algoritmo Descentralizado de Localização**  
**para Rede de Sensores Sem Fio Usando**  
**Calibragem Cooperativa e Heurísticas**

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Ciência da Computação

Orientador: Prof. Dr. Antônio Augusto Fröhlich

Florianópolis, Fevereiro de 2007

**Um Algoritmo Descentralizado de Localização  
para Rede de Sensores Sem Fio Usando  
Calibragem Cooperativa e Heurísticas**

Ricardo Reghelin

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação Área de Concentração Sistemas de Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

---

Prof. Dr. Rogério Cid Bastos

Coordenador do Programa de Pós-Graduação em Ciência da Computação

Banca Examinadora

---

Prof. Dr. Antônio Augusto Fröhlich

Orientador

---

Prof. Dr. Markus Endler

---

Prof. Dr. Mário Antonio Ribeiro Dantas

---

Prof. Dr. João Bosco Manguiera Sobral

## **AGRADECIMENTOS**

Agradeço a Deus por todas as oportunidades, pela saúde, por tudo.

Agradeço aos meus Pais pela oportunidade que me deram de me oferecer um caminho, uma profissão.

Agradeço a minha esposa Luciane, que sempre me apoiou nos devaneios de pesquisa, nas horas de certa dúvida no êxito, nos finais de semana sem passeio, e nas férias que não pudemos viajar.

Meu sincero agradecimento aos professores e em especial ao meu orientador Professor Doutor Antônio Augusto Fröhlich pela paciência, pelos ensinamentos, pelas conversas e esclarecimentos.

Agradeço também aos amigos do LISHA.

Muito obrigado.

## SUMÁRIO

<b>LISTA DAS FIGURAS.....</b>	<b>VII</b>
<b>LISTA DE ABREVIATURAS .....</b>	<b>IX</b>
<b>RESUMO .....</b>	<b>X</b>
<b>ABSTRACT .....</b>	<b>XI</b>
<b>1 INTRODUÇÃO.....</b>	<b>12</b>
1.1 O PROBLEMA.....	12
1.2 JUSTIFICATIVA .....	13
1.3 OBJETIVO .....	14
1.4 HIPÓTESE .....	15
1.5 ESTRUTURA DA DISSERTAÇÃO.....	15
<b>2 SISTEMAS DE LOCALIZAÇÃO EM REDES DE SENSORES SEM FIO .....</b>	<b>17</b>
2.1 ESTIMATIVA DE PROXIMIDADE/DISTÂNCIA/ÂNGULO.....	20
2.1.1 RSSI - Intensidade do sinal recebido.....	21
2.1.2 ToA/TDoA/ToF – medição por cronometragem.....	22
2.1.3 AoA/DoA - Ângulo/direção de chegada.....	23
2.1.4 Conectividade .....	24
2.1.5 Considerações sobre a estimativa de proximidade/distância/ângulo.....	24
2.2 CÁLCULO DE POSIÇÃO .....	25
2.2.1 Trilateração.....	25
2.2.2 Multilateração .....	27
2.2.3 Triangulação .....	28
2.2.4 Bounding box.....	29
2.2.5 Posição central em relação às referências.....	29
2.3 ALGORITMO DE LOCALIZAÇÃO .....	30
2.3.1 ABC - Assumption Based Coordinate.....	31
2.3.2 TERRAIN .....	31
2.3.3 APS - Ad hoc positioning system .....	32
2.3.4 Hop-Terrain.....	33
2.3.5 Recursive Position Estimation (RPE).....	34
2.3.6 MDS – Multidimensional Scaling.....	35
2.3.7 Considerações sobre o algoritmo de localização.....	35
2.4 SUMÁRIO.....	36
<b>3 O ALGORITMO HECOPS.....</b>	<b>38</b>

3.1 MELHORIAS NA MEDIÇÃO DE ALCANCE .....	38
3.1.1 HEURÍSTICA.....	38
3.1.2 FUNÇÃO <i>tri</i> .....	40
3.1.3 DESVIO PERMANENTE.....	42
3.1.4 TODOS OS SENSORES SÃO REFERÊNCIAS.....	43
3.1.5 TRANSMISSÃO EM DIFERENTES NÍVEIS DE POTÊNCIA .....	43
3.2 SISTEMA DE COORDENADAS .....	43
3.2.1 CÁLCULO DA POSIÇÃO .....	44
3.2.2 ESCOLHA DAS REFERÊNCIAS.....	45
3.2.3 ATRIBUINDO VALORES DE CONFIANÇA.....	46
3.3 FUNCIONAMENTO .....	49
3.3.1 TABELA DE DADOS.....	49
3.3.2 ATUALIZAÇÃO DA TABELA DE DADOS .....	50
3.3.3 EXEMPLO.....	51
<b>4 RESULTADOS EXPERIMENTAIS .....</b>	<b>57</b>
4.1 SIMULADOR .....	57
4.2 RESULTADOS .....	58
<b>5 CONCLUSÕES E TRABALHOS FUTUROS.....</b>	<b>64</b>
5.1 CONCLUSÕES.....	64
5.2 TRABALHOS FUTUROS .....	65
<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>66</b>

## LISTA DAS FIGURAS

FIGURA 1: LINHA DE TEMPO DOS SISTEMAS DE LOCALIZAÇÃO. ....	18
FIGURA 2: ETAPAS DE LOCALIZAÇÃO: (A) ASSOCIAR PONTOS DESCONHECIDOS AOS PONTOS DE REFERÊNCIA, (B) CALCULAR A POSIÇÃO USANDO OS PONTOS DE REFERÊNCIA E AS RELAÇÕES PARA ESTE. ....	19
FIGURA 3: TDOA –USANDO-SE DOIS SINAIS COM TEMPOS DE CHEGADA DIFERENTES PODE-SE DETERMINAR A DISTÂNCIA[SAVVIDES 2001].....	23
FIGURA 4: TRILATERAÇÃO: A SOLUÇÃO É A INTERSECÇÃO DE TRÊS CÍRCULOS CENTRADOS NA POSIÇÃO DAS REFERÊNCIAS, TENDO O RAIOS IGUAL A CADA MEDIÇÃO DE ALCANCE.....	26
FIGURA 5: TRILATERAÇÃO CONSIDERANDO ERROS: (A) A ÁREA ESCURA REPRESENTA AS SOLUÇÕES POSSÍVEIS; E (B) SOLUÇÃO DO PROBLEMA PELO VALOR RESIDUAL MÍNIMO.....	26
FIGURA 6: MULTILATERAÇÃO, SEMELHANTE À TRILATERAÇÃO, PORÉM MAIS DE TRÊS REFERÊNCIAS.....	28
FIGURA 7: TRIANGULAÇÃO (A) UTILIZANDO TRÊS REFERÊNCIAS UM SENSOR É CAPAZ DE CALCULAR SUA POSIÇÃO; E (B) DOIS SENSORES PODEM CALCULAR A POSIÇÃO DE UM TERCEIRO SENSOR.....	28
FIGURA 8: <i>BOUNDING BOX</i> . A INTERSECÇÃO DOS QUADRADOS FORMAM UM RETÂNGULO (ESCURO). O CENTRO DESTES É O RESULTADO. ....	29
FIGURA 9: <i>ABC</i> . (A) UM SENSOR INICIA COMO REFERÊNCIA E UM SEGUNDO SENSOR ASSUME POSIÇÃO, (B) UM TERCEIRO SENSOR ASSUME POSIÇÃO BASEADO NOS DOIS PRIMEIROS, (C) COM 3 REFERÊNCIAS O SISTEMA NÃO É MAIS INDETERMINADO. ....	31
FIGURA 10: <i>APS</i> . EXEMPLO DE CÁLCULO DA DISTÂNCIA MÉDIA DE UM SALTO. ENTRE ÂNCORAS L2 E L1 É 20M (40M DIVIDIDO POR DOIS SALTOS) [NICULESCU 2001]. ....	32
FIGURA 11: COMPARAÇÃO DE ALGORITMOS DISTRIBUÍDOS DE LOCALIZAÇÃO. ....	35
FIGURA 12: PADRÃO DE PROPAGAÇÃO IRREGULAR DE RF EM UM SENSOR [JI 2004]. ....	39
FIGURA 13: CONTORNO DAS PROBABILIDADES DE RECEPÇÃO DE PACOTES DE UM SENSOR CENTRAL [GANESAN 2002]. ....	40
FIGURA 14: EXEMPLO DO SENSOR C SENDO AFETADO PELO DESVIO DETECTADO AB; FUNÇÃO <i>TRI</i> (AB) MEDIÇÃO DE ALCANCE AC CALIBRADA PELO SENSOR B. ....	41
FIGURA 15: ESTUDO DO LIMITE MÍNIMO FIXADO PARA CONFIANÇA DE UM SENSOR PARA PARTICIPAR DE UMA FUNÇÃO. ESCOLHIDO VALOR 0,8. ....	47
FIGURA 16: ESTUDO DO PESO (%) ATRIBUÍDO ÀS REFERÊNCIAS NO CÁLCULO DE C. ESCOLHIDO VALOR 75%. .....	48
FIGURA 17: EXEMPLO DE TABELA DE DADOS DE UM SENSOR A QUE ARMAZENA REFERÊNCIAS DE QUATRO OUTROS SENSORES B, C, D E E. ....	50
FIGURA 18: EXEMPLO DE FUNCIONAMENTO DE REDE COM DOIS SENSORES ÂNCORAS A E B E DOIS SENSORES COM POSIÇÃO DESCONHECIDA B E D, TODOS SE COMUNICAM.....	52



FIGURA 19: EXEMPLO DE FUNCIONAMENTO – ESTADO INICIAL, TABELAS VAZIAS EXCETO AS DOS ÂNCORAS. .....	52
FIGURA 20: EXEMPLO DE FUNCIONAMENTO – APÓS TRANSMISSÃO DO SENSOR A, É ADICIONADA UMA LINHA EM CADA TABELA DOS OUTROS SENSORES. ....	53
FIGURA 21: EXEMPLO DE FUNCIONAMENTO – APÓS TRANSMISSÃO DO SENSOR B, SENSOR D AGORA TEM DUAS REFERÊNCIAS E RECALCULA POSIÇÃO E CONFIANÇA. ....	54
FIGURA 22: EXEMPLO DE FUNCIONAMENTO – ANTES DA TRANSMISSÃO DO SENSOR C, SEGUNDO CICLO. ...	54
FIGURA 23: EXEMPLO DE FUNCIONAMENTO – SENSOR D IDENTIFICA A FUNÇÃO $TRI(AC)$ , A ÁREA HACHURADA INDICA A ÁREA AFETADA PELO DESVIO ENTRE A E C. ....	56
FIGURA 24: EXEMPLO DE FUNCIONAMENTO – APÓS TRANSMISSÃO DO SENSOR C, SEGUNDO CICLO, O SENSOR D IDENTIFICA A FUNÇÃO $TRI(AC)$ E ARMAZENA NA LINHA AD DA TABELA. ....	56
FIGURA 25: EXEMPLO DE SITUAÇÕES DA FUNÇÃO $TRI$ : (A) BAIXA PROBABILIDADE DE CALIBRAGEM E (B) ALTA PROBABILIDADE DE CALIBRAGEM.....	58
FIGURA 26: DESEMPENHO DO HECOPS - MÉDIAS DOS ERROS DE POSIÇÃO VARIANDO-SE A CONECTIVIDADE. .....	59
FIGURA 27: DESEMPENHO DO HECOPS - COMPARAÇÃO DA MÉDIA DE ERROS DE POSIÇÃO: HECOPS E HOP- TERRAIN. ....	60
FIGURA 28: DESEMPENHO DO HECOPS - COMPARAÇÃO DE RESULTADOS AO LONGO DE ITERAÇÕES (HECOPS E HOP-TERRAIN). ....	60
FIGURA 29: DESEMPENHO DO HECOPS - MÉDIA DE ERROS DE POSIÇÃO RESTRINGINDO O NÚMERO DE REFERÊNCIAS NA TABELA. ....	61
FIGURA 30: RELAÇÃO ENTRE CONFIANÇA E ERRO DE POSIÇÃO (MÉDIA E DESVIO PADRÃO).....	62
FIGURA 31: DESEMPENHO DO HECOPS EM UM PROBLEMA ESPECÍFICO. ....	63

## LISTA DE ABREVIATURAS

ABC	Assumption Based Coordinate
AoA	Angle of Arrival
APS	Ad hoc positioning system
DGPS	Differential Global Positioning System
DoA	Direction of Arrival
GPS	Global Positioning System
HECOPS	Heuristic Environmental Consideration Over Positioning System
MA	Medição de alcance
MDS	Multidimensional scaling
RPE	Recursive position estimation
RSSF	Rede de sensores sem fio
RSSI	Received Signal Strength Indicator
TDoA	Time difference of arrival
ToA	Time of arrival

## RESUMO

REGHELIN, Ricardo. **Um Algoritmo Descentralizado de Localização para Rede de Sensores Sem Fio Usando Calibragem Cooperativa e Heurísticas**. 2007. 68 f. Dissertação (Mestrado em Ciência da Computação) – Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de Santa Catarina, Florianópolis, 2007.

Este estudo trata do problema de determinar a localização de sensores dentro de uma rede de sensores sem fio (RSSF), descrevendo um algoritmo completamente descentralizado chamado HECOPS, onde cada sensor estima sua própria posição após interagir com outros sensores da rede. Somente uma parcela da população de sensores conhece exatamente sua posição. Entretanto qualquer sensor da rede pode ser escolhido como referência. É proposto um sistema de ranqueamento do nível de confiança na posição estimada de cada sensor. Isto leva a uma nova abordagem do problema onde menos referências são escolhidas, porém mais confiáveis, reduzindo-se assim a propagação de erros e comunicações desnecessárias. São propostas também heurísticas para reduzir os erros na medição de alcance. O principal se refere a um esquema para calibrar as medições de alcance através da comparação, quando possível, da distância estimada com a distância real entre os pares de sensores. Experiências realizadas em um ambiente simulado demonstraram que o algoritmo tem um desempenho superior a um modelo anterior em termos de precisão e robustez.

**Palavras-chave:** Localização em redes de sensores sem fio, algoritmo de localização.

## ABSTRACT

This work studies the problem of determining the location of nodes in a wireless sensor network, describing a fully decentralized algorithm called HECOPS, where every node estimates its own position after interacting with other nodes. Only a limited number of nodes have exact knowledge of their position coordinates. Any node can, however, be selected as a reference. We propose a ranking system to determine the reliability of each estimated position. This leads to a novel approach for position calculation that uses fewer but more reliable landmarks, thus reducing data communication and limiting error propagation. We proposed a heuristics that is used to reduce the effects of measurement errors, including a scheme to calibrate range measurements by comparing, whenever possible, the estimated distance with the actual distance between a pair of nodes. Experiments demonstrate that the algorithm is superior to a previously proposed method in terms of its ability to compute correct coordinates under a wider variety of conditions and its robustness to measurement errors.

**Keywords:** Localization Algorithms, Wireless Sensor Networks

# 1 INTRODUÇÃO

A viabilidade do uso da tecnologia de Redes de Sensores sem Fio (RSSF) tem crescido devido aos avanços na área de micro eletrônica tais como: redução dos custos de manufatura de semicondutores, miniaturização, novos desenhos de circuitos de menor consumo de energia, e equipamentos de transmissão sem fio de baixa potência mais simples, porém confiáveis.

Através da implantação de um grande número de dispositivos de *hardware* de baixo custo, chamados sensores, ou nós, espalhados próximos aos fenômenos de interesse, torna-se possível a coleta de dados e monitoramento de ações. Cada dispositivo sensor é usualmente capaz de sentir um fenômeno específico, processar os dados em pequena escala e comunicar-se através de um rádio transmissor.

## 1.1 O PROBLEMA

Este trabalho de pesquisa trata do problema de localização em uma RSSF. Supondo um grupo de sensores dos quais não são conhecidas as coordenadas de posição e um mecanismo pelo qual um sensor consegue estimar sua distância até outro, deseja-se então determinar a coordenada de posição de cada sensor através da comunicação entre os sensores. Um sensor é considerado conectado se consegue transmitir e receber dados de outros sensores da rede.

Uma grande complicação deste problema é que os dispositivos de localização como o de localização por satélite – GPS (*Global Positioning System*) apresentam várias desvantagens [Niculescu 2001]:

- Alto custo do *hardware*;
- Falta de visada aos satélites, ou seja, não operam em ambientes cobertos;
- Tamanho muito grande para serem instalados e utilizados em grande volume;
- Consumo de energia elevado, o que reduz a vida útil;

- Algumas vezes são imprecisos, necessitando mecanismos adicionais como DGPS (*Differential Global Positioning System*);

Este problema é abordado aqui, como em muitos casos na literatura, equipando-se apenas uma parcela da rede com dispositivos de localização ou mesmo instalando-se manualmente sensores cuja localização é conhecida *a priori*. Estes sensores são chamados âncoras ou “*beacons*”. Todos os outros sensores determinam suas próprias localizações através da interação com sensores âncoras e não âncoras. Esta determinação é o próprio problema de localização de sensores em uma RSSF.

## 1.2 JUSTIFICATIVA

Quando se deseja desenvolver aplicações para esta área de RSSF, rapidamente fica claro que sem o conhecimento da posição dos dispositivos perde-se muito da sua utilidade [Girod 2001].

A definição de um sistema de localização é a parte principal da interface do mundo real com o mundo virtual [Estrin 2002].

As RSSFs têm muitas aplicações interessantes na coleta de dados e rastreamento de objetos em movimento [Lorincz 2004], tais como:

- Observação do meio ambiente e ambiental [Biagioni 2002, Wang 2003, Lorincz 2004];
- Na área da saúde como no monitoramento de médicos e pacientes em um hospital [Akyildiz 2002, Schwiebert 2001];
- Em propósitos militares como rastreamento de inimigos e detecção de materiais agressivos como gases tóxicos, elementos radiativos, explosivos,... [Boettcher 2002, Howden 2002];
- No monitoramento de situações de emergência como catástrofes naturais ou informação de mau funcionamento de equipamento em uma fábrica;
- No monitoramento de itens de estoque em um armazém e tráfego de veículos.

Muitas destas aplicações, entretanto dependem de um sistema apropriado de localização que defina o posicionamento de todos os sensores. Também alguns

protocolos em desenvolvimento baseiam-se na vantagem de saber a localização geográfica dos sensores [Bulusu 2000, Savvides 2001].

No estudo de localização a calibragem individual dos dispositivos é fundamental para atingir precisão de resultados. Entretanto se torna inviável devido ao grande número utilizado e falta de um dispositivo interface adequado [Whitehouse 2002]. Em muitos casos os sensores são simplesmente jogados no ambiente sem conhecimento da topologia. Desta forma é importante estabelecer uma forma dinâmica de calibragem considerando os efeitos do meio ambiente. No algoritmo proposto o sistema possui característica de auto-calibração. É um fator inerente ao seu desenho e que não solicita equipamentos adicionais. Muitos estudos de auto-calibração consideram modelos matemáticos com difícil aplicação em ambiente distribuídos pela limitação de recursos [Slijepcevic 2002].

### **1.3 OBJETIVO**

O objetivo desta dissertação é propor um novo algoritmo de localização de RSSFs que deve apresentar as seguintes vantagens :

- Auto-configurável: o sistema de localização deve funcionar baseado nas coordenadas dos sensores dentro na rede e não depender de suporte de outra infra-estrutura;
- Robusto: o sistema deve considerar as imprecisões nas medições de distância dos sensores e prever a falta de alguns deles;
- Alta precisão: deve reduzir os erros de medição através de um sistema de auto-calibragem;
- Deve estabelecer para cada sensor da rede um valor relativo ao nível de confiança em sua posição, o que pode permitir o desenvolvimento de outras aplicações.

No intuito de alcançar o objetivo geral citado acima, são necessários os seguintes objetivos específicos:

- Pesquisar e analisar os sistemas de medição/localização, os métodos de cálculo aplicados ao problema e os algoritmos de localização;
- Propor um novo algoritmo descrevendo em detalhes seu funcionamento;

- Avaliar o desempenho através de simulações em vários cenários e comparação com pelo menos um algoritmo existente;
- Analisar as limitações e trabalhos futuros que podem utilizar o algoritmo proposto.

## **1.4 HIPÓTESE**

Um sensor pode medir sua distância até outro através de mecanismos de medição de alcance, como por exemplo, o RSSI que é descrito no próximo capítulo. Alguns métodos resultam em erros superiores a 50%. Os erros provêm de muitas fontes, incluindo: interferência entre múltiplos canais, obstáculos na linha de frente e atenuação de sinal [Savvides 2001].

Outra maneira de se obter a distância entre dois sensores é estimá-la através de cálculos baseadas nas informações de todos os sensores da rede. Neste caso também aparecem imprecisão devido principalmente à grande limitação de capacidade computacional [Slijepcevic 2002].

A hipótese básica neste estudo é de que dois sensores quaisquer dentro de uma RSSF, que conhecessem suas próprias posições com certa confiabilidade de precisão, são capazes de detectar diferenças entre as distâncias medidas e distâncias estimadas. Estas eventuais diferenças detectadas podem então ser usadas para melhorar a precisão nos cálculos.

## **1.5 ESTRUTURA DA DISSERTAÇÃO**

O trabalho está organizado em cinco capítulos da seguinte forma:

No próximo capítulo é apresentada a revisão da literatura com as soluções propostas até então. São descritos os três componentes de um sistema de localização: identificação das referências, cálculo da posição e algoritmos.

No capítulo 3 é descrito o funcionamento do algoritmo e a contribuição da heurística, que é a mais importante neste trabalho. Também é apresentado um exemplo de funcionamento mostrando como ocorrem as trocas de informações entre os sensores.

No capítulo 4 são apresentados os experimentos realizados para testar o desempenho do algoritmo proposto. É apresentado o método de simulação utilizado e os



parâmetros que são alterados neste problema. Os resultados do algoritmo são comparados na mesma base com os resultados de um outro algoritmo já existente.

O capítulo 5 encerra o trabalho com as conclusões obtidas e sugestões de trabalhos futuros.

## 2 SISTEMAS DE LOCALIZAÇÃO EM REDES DE SENSORES SEM FIO

Primeiramente, com a finalidade de facilitar o entendimento do trabalho descrito no próximo capítulo, é importante conhecer alguns conceitos relacionados à área de localização em RSSF. Este capítulo é dedicado a esta explanação.

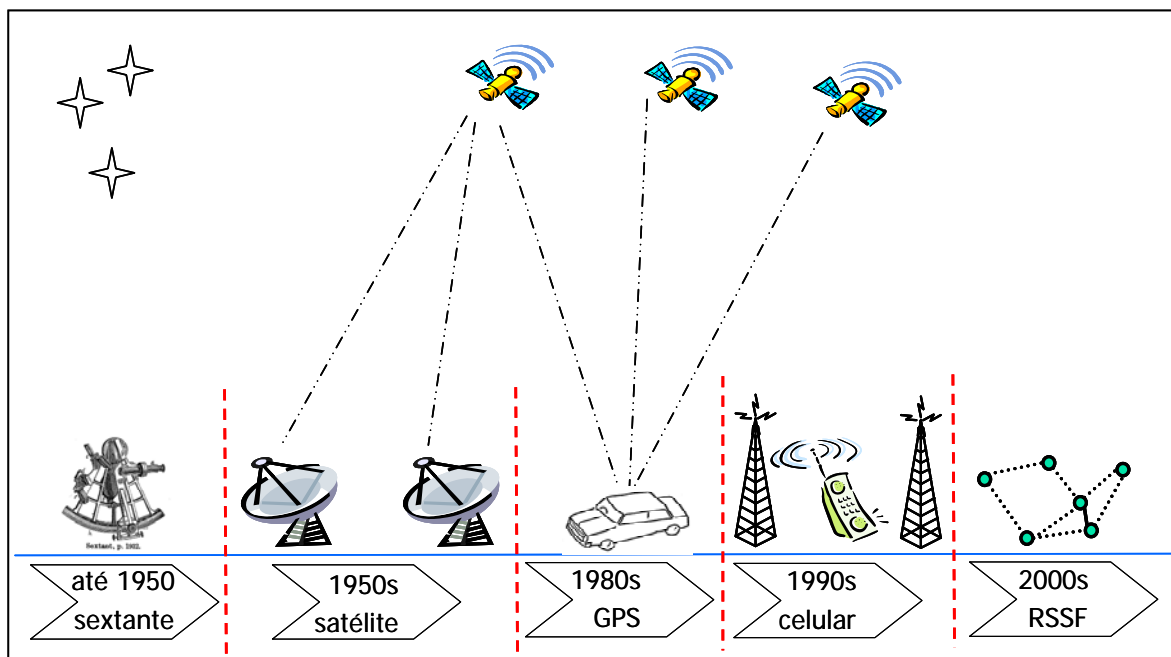
O problema da localização consiste basicamente em identificar a posição física (e.g., latitude, longitude, altitude) de um determinado objeto. Na literatura este problema é estudado de várias formas, abrangendo áreas como redes *ad hoc*, redes de sensores sem fio, telefonia móvel, robótica e militar. Neste trabalho o problema de localização é abordado com relação às RSSFs.

As soluções foram evoluindo e acompanhando as novas tecnologias disponíveis como pode ser observado na Figura 1.

A necessidade de localização é antiga e as primeiras soluções também. Antes de 1950, surgem as primeiras aplicações de localização com as primeiras navegações por terra e ar. Neste período as estrelas eram usadas como pontos de referência. Os ângulos entre as estrelas e o horizonte eram medidos com o uso do sextante. As coordenadas eram calculadas baseadas nestas duas medições.

Após 1950 surge a necessidade de rastrear os satélites no espaço. Neste período as bases terrestres eram usadas como referência e para o cálculo se considerava as distâncias entre o satélite e as bases.

Na década de 80 surge o GPS (*Global Positioning System*). Agora os satélites são usados como referência e as distâncias entre eles e entre eles e o ponto na terra são medidos. As posições dos satélites são conhecidas e, portanto pode-se calcular as coordenadas na superfície da terra.



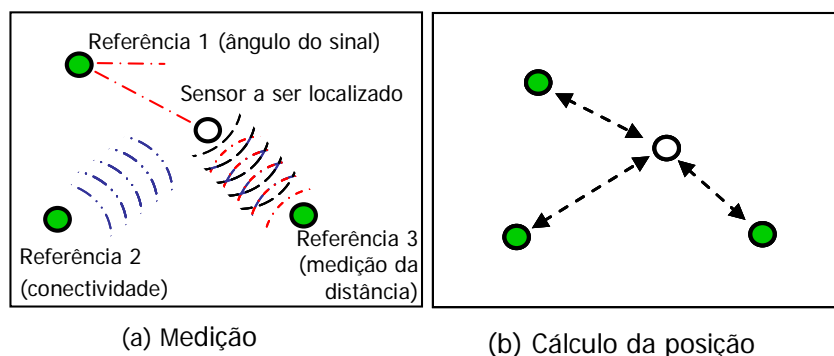
**Figura 1: Linha de tempo dos sistemas de localização.**

No início dos anos 90 surge a telefonia celular e com ela a aplicação de localização de usuários móveis, especialmente útil para os casos de emergência [Caffery 1998]. Neste caso, as estações rádio base são os pontos de referências. A localização dos usuários móveis é obtida primeiramente pela medição de ângulo ou distância até uma estação cuja posição é conhecida, e então, calculam-se as posições.

Finalmente a partir do ano 2000 surgem as redes de sensores sem fio. Neste caso os sensores se utilizam da posição dos sensores vizinhos e das distâncias para estes sensores afim de calcularem as suas próprias posições.

Como se observa na linha de tempo do problema de localização, uma vez que as referências estão disponíveis, o processo de localização se completa em duas etapas [Karalar 2005]:

- 1- Associar os pontos desconhecidos aos pontos de referência;
- 2- calcular a posição através de um algoritmo usando os pontos de referência e as relações para estes.



**Figura 2: Etapas de localização: (a) associar pontos desconhecidos aos pontos de referência, (b) calcular a posição usando os pontos de referência e as relações para este.**

Estas etapas podem ser representadas conforme a Figura 2. Primeiro um sensor com posição desconhecida procura pontos de referência e respectivos relacionamentos que podem ser a distância estimada, medida de ângulo ou simplesmente a proximidade a ela.

Uma abordagem genérica para determinar a posição de um sensor foi proposta por Langendoen. Este estudo identifica três fases nos algoritmos de localização distribuídos [Langendoen 2003]:

1. determinação das distâncias entre sensores âncoras e não âncoras;
2. estimativa das posições dos sensores em relação aos âncoras;
3. refinamento das estimativas de posição baseado nas medições das distâncias (MA) e posições dos sensores vizinhos.

Em um estudo de Hightower pode ser encontrado um estudo mostrando várias tecnologias de localização, incluindo um comparativo de 15 sistemas de localização relacionando precisão, custo e limitações [Hightower 2001].

Os sistemas de localização estão divididos em três componentes [Oliveira 2005]:

1. Estimativa de proximidade/distância/ângulo: este componente é responsável por estimar informações de proximidade, distâncias e/ou ângulos entre todos os sensores, referências e não-referências. Ele fornece as informações para os outros componentes do sistema de localização.

2. Cálculo de posição: este componente é responsável por calcular a posição do sensor baseado nas informações recebidas do componente anterior.

3. Algoritmo de localização: é o componente necessário para disponibilizar referências adicionais quando as informações para o cálculo são insuficientes ou

imprecisas. Por exemplo, algumas vezes as referências não estão disponíveis diretamente em sensores adjacentes e exigem estratégias de multi-salto entre os sensores.

Importante ressaltar que estes três componentes contribuem de forma diferente para o objetivo final, que é a determinação da posição. Desta forma são estudados separadamente na literatura, e assim serão apresentadas neste trabalho.

Nas próximas seções deste capítulo os componentes são detalhados e analisados os principais recursos empregados por cada um deles. No final do capítulo uma conclusão geral do assunto é realizada.

Para simplificar a apresentação e análise da teoria, são considerados apenas duas dimensões na área de RSSF.

## **2.1 ESTIMATIVA DE PROXIMIDADE/DISTÂNCIA/ÂNGULO**

A primeira etapa do processo de localização é fundamental e diz respeito à identificação de pontos de referência e outros sensores vizinhos. O resultado desta identificação pode se apresentar basicamente de três formas:

- Proximidade: pela detecção da presença de um sensor vizinho;
- Medição da distância até ele usando dispositivo específico para este fim;
- Medição do ângulo entre os sensores.

Existem diferentes métodos para obtenção de tais informações. A escolha do método depende principalmente dos custos e dimensional do dispositivo. Existe uma relação direta entre precisão de resultados e custo do dispositivo [Hightower 2001]. Dentre outros fatores de custo, pode-se citar o consumo de energia e nível precisão necessários.

Nas próximas seções são apresentados os principais métodos utilizados pelos sistemas de localização para estimativas de proximidade/distância/ângulo: o RSSI, o ToA/TDoA, o AoA e o raio de alcance. Ao final são feitos alguns comentários resumindo os métodos.

### 2.1.1 RSSI - Intensidade do sinal recebido

A intensidade do sinal recebido – RSSI (*Received Signal Strength Indicator*) é um método que estima a distância entre dois sensores baseado na potência do sinal recebido no sensor receptor. O sensor transmissor envia um sinal com uma determinada potência que vai se reduzindo à medida que o sinal se propaga. Quanto maior a distância do sensor receptor, menor resulta a potência do sinal ao atingir este sensor.

A escolha deste método depende de suas vantagens e desvantagens. A principal vantagem é o seu baixo custo comparado com os demais métodos, uma vez que a grande maioria dos receptores é capaz de identificar a potência do sinal recebido. A desvantagem do método está relacionada aos ruídos e interferências que resultam em grande imprecisão nas estimativas de distância. Este é motivo por não ser ainda muito explorado na prática.

Existem praticamente duas técnicas de localização usando RSSI [Vandenbussche 2005]:

- Convertendo sinal em distância. De acordo com a teoria, esta potência diminui com o quadrado da distância. A potência do sinal é comumente medida em dBm (Decibel em referência a um milliwatt) ou mesmo Watts. Alguns experimentos [Savvides 2001] apontam erros de 2 a 3m em cenários onde todos os sensores estão em um campo plano, a uma altura de 1,5 m, e com raio de comunicação de 10 m.
- Impressão digital (“*fingerprinting*”) como, por exemplo, mostrado em [Bahl 2000]. Ela se baseia no comportamento das ondas de rádio em um ambiente específico incluindo todos os efeitos inerentes como reflexão e atenuação de sinal. Seu funcionamento consiste em duas fases. Na primeira fase são coletadas e armazenadas medições do sinal em vários pontos escolhidos. Para cada ponto um sensor recebe sinais de vários sensores emissores, formando uma impressão digital deste ponto. Todas as impressões são armazenadas em um banco de dados. Na segunda fase do algoritmo, um sensor que deseja se localizar coleta os sinais no seu ponto montando uma impressão digital. Esta impressão é então comparada com o banco de dados e a que mais se aproxima indica sua respectiva posição. Esta técnica tem a vantagem de ser mais precisa visto que considera as condições ambientais. Segundo Vandenbussche chega a um metro de precisão. Sua desvantagem é a necessidade de levantar os dados na primeira fase [Vandenbussche 2005].

### 2.1.2 ToA/TDoA/ToF – medição por cronometragem

Nesta categoria estão os métodos que tentam medir o tempo que o sinal percorre a distância que se deseja medir. Existem alguns tipos, mas o mais simples e intuitivo é o ToA (tempo de chegada - *time of arrival*). Neste caso, a distância entre dois pontos é diretamente proporcional ao tempo que o sinal leva para se propagar de um ponto ao outro. Matematicamente pode-se calcular pela seguinte fórmula:

$$d = s_r (t_2 - t_1) \quad (1)$$

onde:

$s_r$  é a velocidade de propagação do sinal de rádio (velocidade da luz)

$t_1$  e  $t_2$  os tempos de saída e chegada do sinal

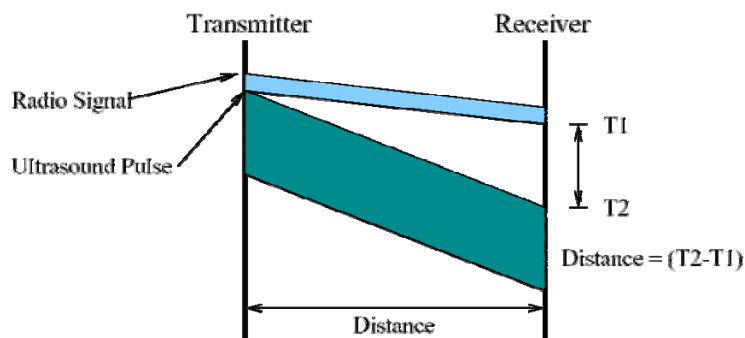
Embora o conceito físico deste método seja extremamente simples, na prática se torna complicado devido à precisão necessária para sincronismo dos relógios do sistema. Este método é muito utilizado, como por exemplo pelo sistema GPS.

O TDoA (diferença do tempo de chegada - *time difference of arrival*) se baseia na diferença de tempo que um sinal chega a múltiplos sensores ou na diferença de tempo que múltiplos sinais chegam a um mesmo sensor. Os sensores que recebem o sinal precisam estar sincronizados. Em redes de sensores é mais comum encontrar dois tipos de sinais sendo transmitidos, portanto os sensores precisam de um dispositivo capaz de enviar/receber dois tipos de sinais simultaneamente. Para possibilitar a medição os dois sinais devem possuir velocidades diferentes de propagação, como por exemplo, rádio/ultra-som ou rádio/acústico. Em geral, o segundo sinal é algum tipo de som, por se propagar mais lentamente.

Savvides *et al.* apresentaram um exemplo de TDoA utilizando um pulso de ultra-som enviado simultaneamente com um sinal de rádio. Neste caso, os sensores medem a diferença entre o tempo de chegada dos dois sinais, como visto na Figura 3 [Savvides 2001]. A distância pode ser então calculada pela fórmula:

$$d = (s_r - s_s) * (t_2 - t_1) \quad (2)$$

Onde  $s_r$  e  $s_s$  são as velocidades de propagação do sinal de rádio e do ultra-som, respectivamente e  $t_1$  e  $t_2$  o tempo de chegada do sinal de rádio e do sinal de ultra-som, respectivamente.



**Figura 3: Tdoa –usando-se dois sinais com tempos de chegada diferentes pode-se determinar a distância[Savvides 2001].**

Conforme testes com ultra-som realizados por Savvides, os erros resultaram em torno de 2 ou 3cm sendo que os sensores estavam separados a uma distância de 3m [Savvides 2001].

Em outro estudo, Whitehouse utilizou som audível (acústico) e resultou em erros um pouco maiores, de 23cm, com sensores separados a uma distância de 2m [Whitehouse 2002].

Uma grande desvantagem deste sistema é a necessidade de *hardware* adicional para o envio e recepção do segundo sinal, o que aumenta o custo do sensor. Outra desvantagem é o raio de alcance do segundo sinal que é geralmente baixo, na faixa dos 3m.

### 2.1.3 AoA/DoA - Ângulo/direção de chegada

Estes métodos utilizam o ângulo de chegada do sinal para determinar sua posição. Este ângulo pode ser em relação ao sensor ou em relação a um segundo sinal recebido também pode ser utilizado pelos sistemas de localização. Exige antenas complexas que, por seu alto custo, são geralmente apenas instaladas nos sensores âncoras ou em estações base.



A medida do ângulo de chegada do sinal é realizada com o auxílio de um conjunto de receptores - em geral três ou mais - separados uniformemente. O cálculo da posição é baseado no tempo de chegada do sinal a cada um dos receptores.

Este método obtém erros de alguns graus (aproximadamente de 5' em Priyantha *et al.* (2001)). A necessidade de *hardware* extra, bem como uma distância mínima entre os receptores, gera desvantagens em termos de custo e tamanho dos sensores.

#### **2.1.4 Conectividade**

Nesta classificação estão os métodos que usam apenas a informação de ter identificado um sensor em seu raio de atuação. Se um sensor recebe um sinal de outro sensor, então a distância entre estes dois está entre zero e o alcance máximo do sinal. As estimativas de posição são do tipo centróide, ou seja, tentam analisar o conjunto de sensores no qual está inserido e daí estimar a sua posição.

Este método de estimativa de distância tem a vantagem de ser o mais simples e menos custoso de todos. Nenhum *hardware* exclusivo e nem cálculos extras são necessários. Por outro lado, um erro de metade do alcance do sinal na estimativa de distância não é viável para a maioria dos sistemas de localização. Considere-se, por exemplo, um raio de alcance de 100m. Neste caso, o erro pode chegar a 50m por estimativa.

#### **2.1.5 Considerações sobre a estimativa de proximidade/distância/ângulo**

Além dos métodos citados, existem outros métodos para estimar as distâncias, como o da luz infravermelha. No uso de luz cada sensor emite em um determinado intervalo de tempo. Devido ao curto alcance, o infravermelho exige o uso de um grande número de sensores para conseguir uma precisão significativa. Além disso, possui problemas de multi-caminhos onde uma recepção esperada de um sensor pode estar vindo de outro. Outro problema é a linha de vista que pode ser interrompida.

## 2.2 CÁLCULO DE POSIÇÃO

Quando um sensor dispuser de informações mínimas relativas aos pontos de referências, pode-se iniciar o cálculo da sua posição empregando um método matemático. A seguir serão apresentados alguns dos mais importantes métodos utilizados nesta área que são: a trilateração, a multilateração, a triangulação, as abordagens probabilísticas, o *bounding box* e a posição central. Deve se ter em mente que a distância entre sensores pode ser estimada utilizando os métodos explicados na seção anterior, e que podem tirar proveito dos pacotes trocados para enviar também a posição conhecida dos sensores referências.

A escolha do método utilizado depende de dois fatores principais que estão relacionados. Quanto mais precisão nos resultados, maior a capacidade de processamento necessária. Outros fatores seriam: o consumo de energia e quantidade de sensores e conseqüentemente de informação a ser tratada.

### 2.2.1 Trilateração

Utilizado para a solução de GPS, a trilateração é o método mais básico e intuitivo. O modelo pode ser utilizado quando são conhecidas as posições de três referências e as distâncias destas referências até o ponto que se deseja estimar. As referências são representadas graficamente por círculos centrados nas respectivas posições e com raio igual à distância até o ponto que se deseja calcular. O resultado está na intersecção de três círculos, conforme ilustrado na Figura 4.

Pela representação gráfica das referências pode se obter a seguinte fórmula:

$$(x-x_i)^2 + (y-y_i)^2 = d_i^2 \quad (3)$$

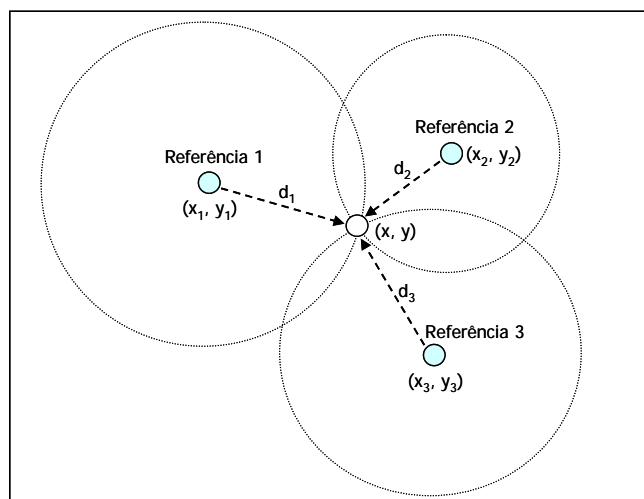
Onde:

$(x,y)$  é a posição que se deseja calcular;

$(x_i,y_i)$  é a posição da referência  $i$  ;

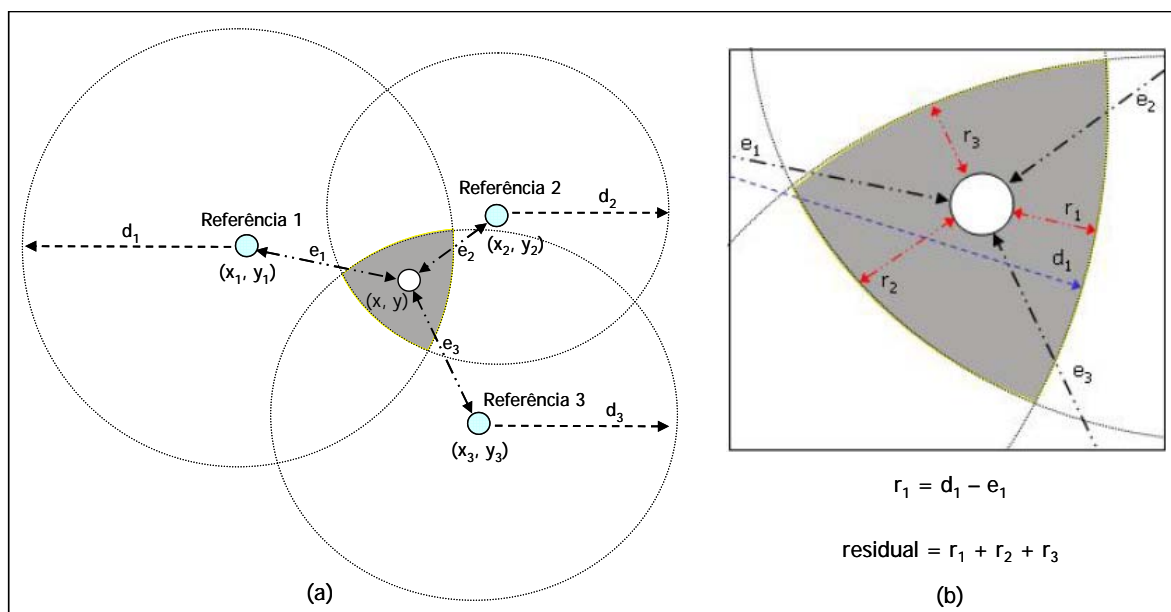
$d_i$  a distância do sensor a referência  $i$ .

Na trilateração têm-se então três equações quadráticas com duas variáveis desconhecidas. Trata-se de um sistema linear de equações que pode ser resolvido por vários métodos, como por exemplo, o método de eliminação de Gauss.



**Figura 4: Trilateração: a solução é a intersecção de três círculos centrados na posição das referências, tendo o raio igual a cada medição de alcance.**

Devido às imprecisões nos dispositivos de medição e fenômenos de propagação de sinais, no mundo real dificilmente a solução gráfica resulta somente em um ponto, mas sim em um conjunto de possíveis soluções como pode ser visto na Figura 5(a) [Oliveira 2005]. Desta maneira, o sistema de equações fica com mais variáveis e demanda métodos mais complexos para a solução.



**Figura 5: Trilateração considerando erros: (a) a área escura representa as soluções possíveis; e (b) solução do problema pelo valor residual mínimo.**

Um dos métodos mais utilizados na literatura é o método dos quadrados mínimos. Como pode ser observado na Figura 5(b) existe uma diferença  $r_i$  entre a distância medida  $d_i$  até a referência e a distância  $e_i$  calculada da posição estimada. Estas diferenças de distâncias são chamadas de valor residual. Este método procura minimizar o quadrado da diferença destas distâncias.

Pode-se representar matematicamente esta função por:

$$f(pos) = \min \left( \sum_{i=1}^m (\|pos - ref_i\| - d_i)^2 \right) \quad (4)$$

Onde:

$pos$  é a posição que se quer calcular,

$ref_i$  é a posição da referência  $i$ ,

$d_i$  a distância medida.

O fator  $\|pos - ref_i\|$  representa a distância entre a posição calculada e a posição da referência  $i$ .

Outros métodos utilizados na literatura incluem a linearização de Taylor e o método de Gauss. A escolha do método dependerá da capacidade computacional e precisão da solução desejada.

### 2.2.2 Multilateração

A multilateração nada mais é que a trilateração, porém considerando mais de três referências para o cálculo da posição. Os princípios matemáticos são semelhantes. A diferença, entretanto, é que se obtém um sistema linear sobredeterminado, pois existem mais equações do que variáveis, como mostra a Figura 6.

Os métodos para a solução são os mesmos citados para a trilateração.

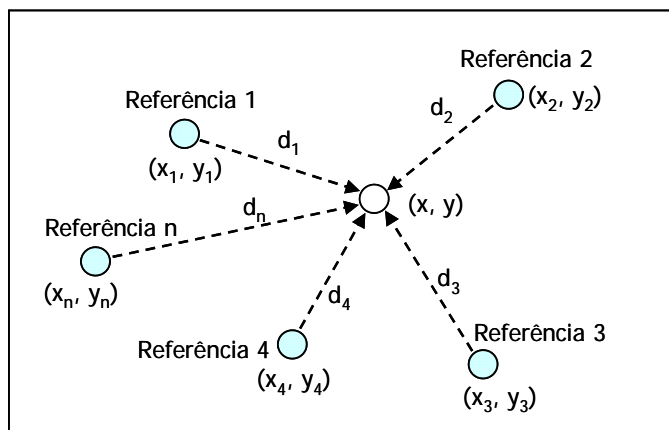


Figura 6: Multilateração, semelhante à trilateração, porém mais de três referências.

### 2.2.3 Triangulação

Na triangulação são usadas informações de ângulos ao invés de distâncias.

Se um sensor puder estimar seu ângulo em relação a três referências, então ele é capaz de calcular sua posição através de relações trigonométricas, como mostra a Figura 7(a).

Outra forma de triangulação é mostrada na Figura 7(b). Neste caso dois sensores estimam o ângulo entre uma referência até um terceiro sensor que para o qual se deseja determinar a posição. Estas informações de ângulo são trocadas entre eles possibilitando o cálculo da posição. Importante notar é que neste caso não é o sensor que calcula sua posição.

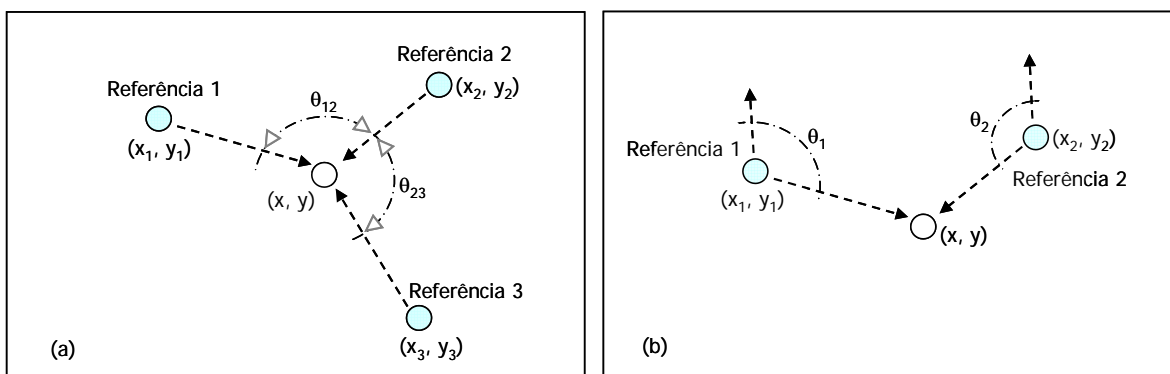
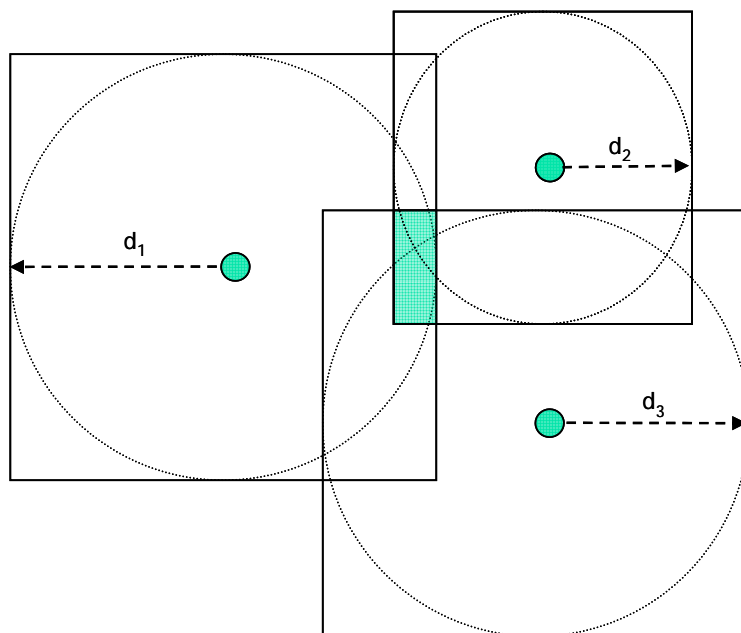


Figura 7: Triangulação (a) utilizando três referências um sensor é capaz de calcular sua posição; e (b) dois sensores podem calcular a posição de um terceiro sensor.

### 2.2.4 Bounding box

O *bounding box*, ou *min-max*, proposto por Savvides et. al, é um esquema que considera quadrados teórico no lugar dos círculos como ocorre na multilateração. Esta aproximação reduz drasticamente os cálculos necessários, embora não tenha uma boa precisão de resultado [Savvides 2002]. Um exemplo do esquema pode ser visto na Figura 8.

Para cada sensor referência  $i$ , um *bounding box* é definido como sendo um quadrado de centro na posição deste sensor  $(x_i, y_i)$  e com lados de tamanho  $2d_i$ . A intersecção de todos os *bounding boxes* pode ser calculada facilmente sem a necessidade de operações de ponto flutuante, formando um retângulo. A posição desejada é então calculada como sendo o centro do retângulo.



**Figura 8: Bounding box. A intersecção dos quadrados formam um retângulo (escuro). O centro deste é o resultado.**

### 2.2.5 Posição central em relação às referências

Uma maneira simplista para se calcular a posição de um sensor é considerá-lo no centro de todos os seus pontos de referência. Desta maneira não é necessário obter informações de distâncias ou ângulos.

Neste caso, a posição de um sensor é calculada através da seguinte fórmula:

$$(x, y) = \left( \frac{\sum_{i=1}^n x_i}{n}, \frac{\sum_{i=1}^n y_i}{n} \right) \quad (5)$$

Onde:  $x$  e  $y$  são as coordenadas de posição.

A grande vantagem deste método é a simplicidade computacional e a de não necessitar de muitas informações. Apenas  $2n + 2$  operações de ponto flutuante (onde  $n$  é o número de sensores) são necessárias para estimar uma posição. A desvantagem é a falta de precisão, diretamente proporcional ao número de referências.

### 2.3 ALGORITMO DE LOCALIZAÇÃO

Devido à natureza de uma RSSF, o algoritmo de localização se torna o principal componente de um sistema de localização. Normalmente os sensores não dispõem de informações suficientes para estimar sua posição baseados apenas em seus vizinhos. Portanto, este componente define uma estratégia de comunicação de toda a rede, obtendo informações antes não consideradas.

Os algoritmos de localização podem ser classificados de várias maneiras, porém serão citados aqueles que ajudarão a focar o estudo e alcançar o objetivo:

- Centralizados ou distribuídos: as posições dos sensores podem ser calculadas de forma centralizada em um sensor ou servidor ou de forma distribuída pelos sensores da rede;
- Com ou sem infra-estrutura: se há a necessidade de criar alguma infra-estrutura prévia para a localização (e.g., posicionamento manual de âncoras);
- Posicionamento relativo ou absoluto: as posições podem estar em coordenadas globais (e.g., latitude, longitude) ou em relação a um sensor ou ponto da rede;
- Um ou múltiplos saltos: se os sensores dependem apenas de seus vizinhos para estimarem a suas posições, ou não.

Contudo, ressalta-se novamente, que o foco deste trabalho está em algoritmos distribuídos. Nas próximas seções, serão, portanto, apresentados os seguintes algoritmos de localização propostos na literatura: o ABC, o TERRAIN, o APS, o Hop-Terrain, o *Recursive position estimation* e o MDS. Ao final será feito um resumo comparativo.

### 2.3.1 ABC - Assumption Based Coordinate

O algoritmo ABC (“*Assumption Based Coordinate*”) proposto por Savarese *et al.* não usa âncoras para encontrar a posição [Savarese 2001]. Um sensor é escolhido como ponto inicial de coordenadas zero. Para os primeiros dois sensores vizinhos ao sensor inicial que recebe sua transmissão, são assumidas coordenadas calculadas por fórmulas propostas. As fórmulas satisfazem a condição geométrica das posições e medição de alcance entre os sensores:

$$n_0 = (0,0) \quad (6)$$

$$n_1 = (r_{01}, 0) \quad (7)$$

$$n_2 = (r_{01}^2 + r_{02}^2 + r_{12}^2) / 2r_{02}, \sqrt{r_{02}^2 - x_2^2} \quad (8)$$

Onde:  $n$  é a posição do sensor;  $r$  a medida de alcance que separa dois sensores.

Com três sensores o sistema não é mais indeterminado e os demais sensores podem calcular suas respectivas posições usando algoritmos comuns (Figura 9).

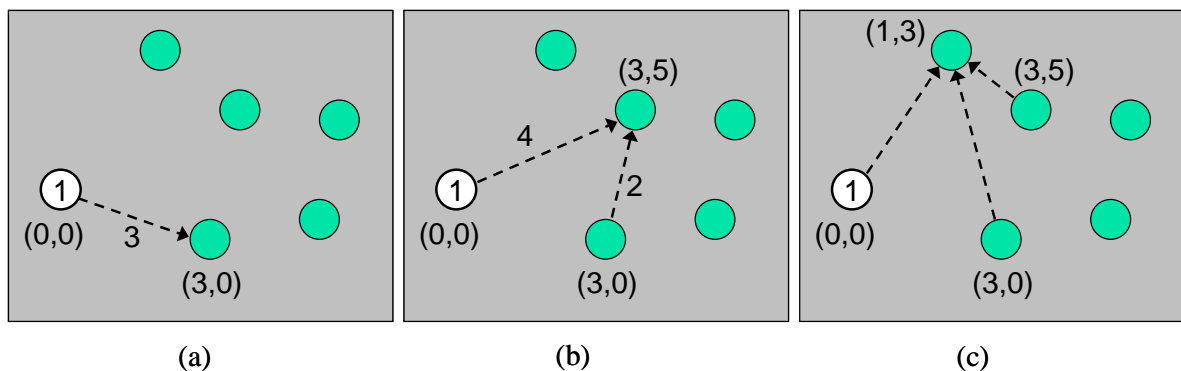


Figura 9: ABC. (a) um sensor inicia como referência e um segundo sensor assume posição, (b) um terceiro sensor assume posição baseado nos dois primeiros, (c) com 3 referências o sistema não é mais indeterminado.

### 2.3.2 TERRAIN

O algoritmo TERRAIN (“*Triangulation via Extended Range and Redundant Association of Intermediate Nodes*”) proposto por Savarese *et al.* é uma extensão do algoritmo ABC [Savarese 2001]. A idéia é que os âncoras iniciem com o ABC inundando a rede. Cada sensor aguarda até receber coordenadas provenientes de ao menos quatro diferentes âncoras. O sensor pode então calcular sua posição por triangulação. À medida que o sensor recebe informações de um número maior de âncoras, a precisão do cálculo de sua posição também aumenta. Também é possível



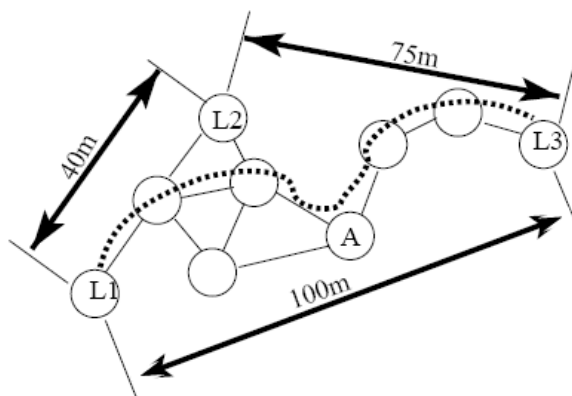
umentar a precisão com um processo iterativo de refinamento, onde novas medições de alcance e posições estimadas mais recentemente pelos sensores vizinhos são usados para recalculer a posição dos sensores. Este processo ocorre até as posições de todos os sensores da rede convergirem.

### 2.3.3 APS - Ad hoc positioning system

Niculescu e Nath propuseram um algoritmo distribuído onde apenas três sensores âncoras são necessários em toda a rede [Niculescu 2001]. A idéia é cada sensor primeiramente estimar sua posição relativa em relação a estes âncoras. Depois os sensores podem calcular suas posições através de multilateração utilizando uma de três maneiras propostas: Dv-hop, Dv-distance e Euclidiano.

No Dv-hop, os âncoras iniciam a propagação de suas informações de localização e todos os outros sensores armazenam as informações de cada âncora, ou seja, sua posição e número de saltos até ele.

Quando um âncora recebe informações de localização de outro âncora, depois desta informação ter saltado por vários sensores, este é capaz de calcular o tamanho médio de um salto, dividindo-se a distância euclidiana pelo número de saltos entre os dois (Figura 10).



**Figura 10:** APS. Exemplo de cálculo da distância média de um salto. Entre âncoras L2 e L1 é 20m (40m dividido por dois saltos) [Niculescu 2001].

Este valor do salto médio é então repassado para a rede, na forma de um fator de correção. A correção partindo dos âncoras é controlada e deve influenciar apenas os sensores mais próximos, ou seja cada sensor recebe preferencialmente somente uma correção. O sensor, ao receber um fator de correção é capaz de transformar a sua

distância que estava em saltos, para metros, podendo então, calcular sua posição usando o método de multilateração.

O Dv-distance funciona de forma semelhante ao Dv-hop. Porém, ao invés de se considerar o número de saltos, considera-se a soma das distâncias estimadas entre os sensores.

O método Euclidiano funciona propagando a distância euclidiana real dos sensores em relação aos âncoras. Para isso, um sensor precisa de pelo menos outros dois sensores que possuam suas distâncias de um determinado âncora, precisa de sua distância a estes dois últimos sensores e também da distância entre os dois sensores. O cálculo da distância do sensor ao âncora é realizado utilizando o teorema de Pitágoras nos triângulos gerados entre as retas formadas pelas distâncias.

Algumas vantagens do APS são a sua simplicidade e a necessidade de poucos sensores âncoras para funcionar. No caso do DV-hop outra vantagem é que o algoritmo não acumula erros de medição de alcance entre os sensores. Porém, as formas como as distâncias são propagadas, em especial o Dv-hop e o Dv-distance geram informações inconsistentes que são utilizadas no cálculo das posições, o que resulta em um sistema com erros altos de localização.

#### **2.3.4 Hop-Terrain**

Este algoritmo proposto por Savarese *et al.* é composto por duas fases. A primeira fase, chamada Hop-Terrain é similar ao DV-hop demonstrado na seção anterior [Savarese 2002].

A segunda fase, chamada refinamento (“*refinement*”), parte de um estado onde todos os sensores já possuem uma estimativa de posição. É um processo em regime permanente buscando mais precisão nos resultados a cada iteração. Neste modo, um sensor interage apenas com seus vizinhos, ou seja, a um salto de distância, e a cada iteração recalcula sua posição baseado nas informações obtidas na interação anterior. Nesta fase a tarefa dos sensores Âncoras é simplesmente transmitir sua posição assim que, em cada nova interação, detectar um novo sensor na rede. A tarefa dos demais sensores é mais complexa. Caso receberem nova informação do processo iterativo anterior, devem calcular nova posição por triangulação. Também devem associar um

nível de confiança associado a esta nova posição estimada. No final ainda devem decidir se devem ou não transmitir estas novas informações para seus vizinhos.

O valor da confiança na posição varia de zero a um. Os âncoras têm o valor máximo igual a um. Sensores com posição desconhecida iniciam com valor igual a 0,1 e podem aumentar ao longo do processo iterativo. Toda a vez que um sensor tiver sucesso e conseguir um resultado de uma triangulação, seu nível de confiança é ajustado para a média de confiança dos seus vizinhos. Desta forma, o valor de confiança aumenta nos sensores próximos aos âncoras e diminui na medida que ficam a alguns saltos de distância dos âncoras.

Caso um sensor falhe na tentativa de convergir um resultado de triangulação, é atribuído zero ao seu valor de confiança. Desta forma seus vizinhos não utilizarão esta informação inconsistente. No algoritmo este valor pode ser decrescido pelo fato de um sensor estar a mais de dois saltos de distância de um âncora ou ainda porque o resultado do cálculo de posição não convergiu, indicando uma elevada incerteza nas informações dos vizinhos. Para melhorar a estimativa de cálculo, o algoritmo também descarta sensores com confianças menores que 0,1 e também limita o número máximo de sensores usados no cálculo.

O algoritmo usa algumas regras para evitar a inundação de informação desnecessária na rede.

### **2.3.5 Recursive Position Estimation (RPE)**

O algoritmo *Recursive position estimation* (RPE) funciona semelhantemente a fase de refinamento do hop-Terrain, ou seja, os sensores estimam suas posições baseados apenas nas informações de seus vizinhos imediatos [Albowicz 2001]. Neste algoritmo cada sensor escolhe quatro ou mais referências dentre os sensores vizinhos e então estima sua posição. Passam então a disseminar sua posição também apenas aos seus vizinhos imediatos. A precisão dos resultados melhora a cada interação do sensor com seus vizinhos.

Este algoritmo inova atribuindo um nível de confiança a cada sensor. O valor é baseado no residual do cálculo de posição. Desta forma, as referências menos confiáveis, ou seja, que não convergiram no resultado matemático, não participam no cálculo de seus vizinhos.

Uma vantagem deste algoritmo é a forma rápida com que os sensores calculam suas posições. Entretanto a desvantagem é a propagação de erro de localização, de modo que quanto mais longe um sensor estiver das referências, maior será seu erro de localização. Além disso, o algoritmo é inteiramente distribuído, não necessita de infraestrutura, estabelece posicionamento absoluto, mais apropriado para ambientes externos e opera com multi-salto.

### 2.3.6 MDS – Multidimensional Scaling

O algoritmo proposto por Ji e Zha usa técnica chamada *multidimensional scaling* (MDS) para aproximar os dados estimados dos reais [Ji 2004]. Neste método, um único sensor (*starting node*) inicialmente inunda a rede até se comunicar com 3 ou mais âncoras chamados *ending anchors*. Os *ending anchors* enviam então de volta para o *starting node* sua posição e a rota dos sensores até ele. O *starting node* calcula então sua posição por multilateração baseado nas distâncias (acumuladas entre os sensores da rota) aos *ending anchors*. Através do processo de MDS, o algoritmo ajusta as posições dos sensores na respectiva rota. Em outras palavras, ajustando a posição estimada do *ending anchor* à posição real do *ending anchor*, o algoritmo afere também, na mesma proporção, a posição de todos os sensores desta rota.

Este conceito provavelmente é o que mais se aproxima do conceito proposto neste trabalho, que é o de identificar um desvio entre o valor estimado e o valor real.

### 2.3.7 Considerações sobre o algoritmo de localização

Resumindo os algoritmos citados nesta seção, na Figura 11 pode-se ver a comparação dos algoritmos com respeito à necessidade de âncora, número de referências e se existe um sistema de ranqueamento de confiança na posição estimada.

algoritmo	necessidade de âncoras	número de referências necessárias	atribuição de nível de confiança ao sensor	processo de refinamento
ABC	não	zero	não usa	não
Terrain	não	4 ou mais	não usa	sim
APS	sim	3 ou mais	não usa	sim
Hop-terrain	não	3 ou mais	média dos vizinhos	sim, na segunda fase
RPE	não	4 ou mais	resíduo de fórmula	sim
MDS	sim	3 ou mais	não	sim

Figura 11: Comparação de algoritmos distribuídos de localização.

A escolha do tipo do algoritmo depende basicamente da aplicação, dos recursos disponíveis, da topologia da rede e da precisão desejada. Para se elaborar uma proposta de um novo algoritmo, que é o objetivo desta dissertação, dois fatores principais foram observados deste comparativo: o conceito de confiança e número de referências necessárias.

O conceito de nível de confiança utilizado no algoritmo Hop-Terrain se mostrou adequado, porém conforme reportado pelo autor, não resultou em uma relação direta e consistente entre nível de confiança e precisão no posicionamento.

Com relação ao número de referências, todos os algoritmos usaram a estratégia de utilizar o maior número possível de referências para o cálculo de posição. Tentando desta forma aumentar a precisão. Embora no Hop-Terrain se tentasse filtrar algumas referências contendo erros, a conclusão é que ocorre propagação de erros em todos os casos.

O processo de refinamento, onde a cada interação se recalcula a posição com valores mais precisos, parece ser uma estratégia eficaz e aparece em praticamente todos os algoritmos.

## **2.4 SUMÁRIO**

Neste capítulo o problema de localização foi abordado sob o ponto de vista das redes de sensores sem fio. Os sistemas de localização foram divididos em três componentes: estimativa de proximidade/distância/ângulo, cálculo de posição e algoritmo de localização. Estes três componentes abrem três frentes diferentes e independentes para o estudo de localização.

Os estudos até agora não encontraram, e provavelmente não exista, uma solução única capaz de atender de forma ótima os diversos requisitos das redes de sensores nos mais variados cenários e aplicações. Atualmente a solução parece estar na adequação caso a caso das situações e recursos eventualmente disponíveis.

Concluindo aqui a revisão da literatura, no próximo capítulo é proposta uma nova solução para o problema de localização. Apesar desta proposta não ser confrontada com todos os algoritmos citados neste capítulo, ela obteve resultados superiores em cenários específicos frente ao algoritmo Hop-Terrain. A escolha deste algoritmo se deu ao estilo

de sua simplicidade, popularidade e principalmente por apresentar uma fase de refinamento de dados semelhante ao algoritmo proposto.

### 3 O ALGORITMO HECOPS

Neste capítulo é descrito o algoritmo de localização proposto neste trabalho que foi chamado HECOPS (*Heuristic Environmental Consideration Over Positioning System*). Este algoritmo procura eliminar ou minimizar grande parte das desvantagens dos algoritmos de localização apresentados no capítulo anterior. Para isto, ele trata principalmente os erros de medição através do uso de heurísticas. A mais importante se refere a um esquema para calibrar as medições de alcance através da comparação, quando possível, da distância estimada com a distância real entre os pares de sensores. Este esquema somente é possível porque é assumida uma certa regularidade no comportamento da propagação de rádio frequência do elemento transmissor.

Outro conceito utilizado é o de usar um número menor de referências no cálculo de localização, porém selecionando as de maior nível de confiabilidade. A confiabilidade de uma referência é relativa à confiabilidade da sua posição e depende das informações disponíveis de outros sensores, da configuração da rede e da topologia. O uso da heurística na classificação da confiabilidade é outra inovação deste algoritmo e será descrito em detalhes neste capítulo.

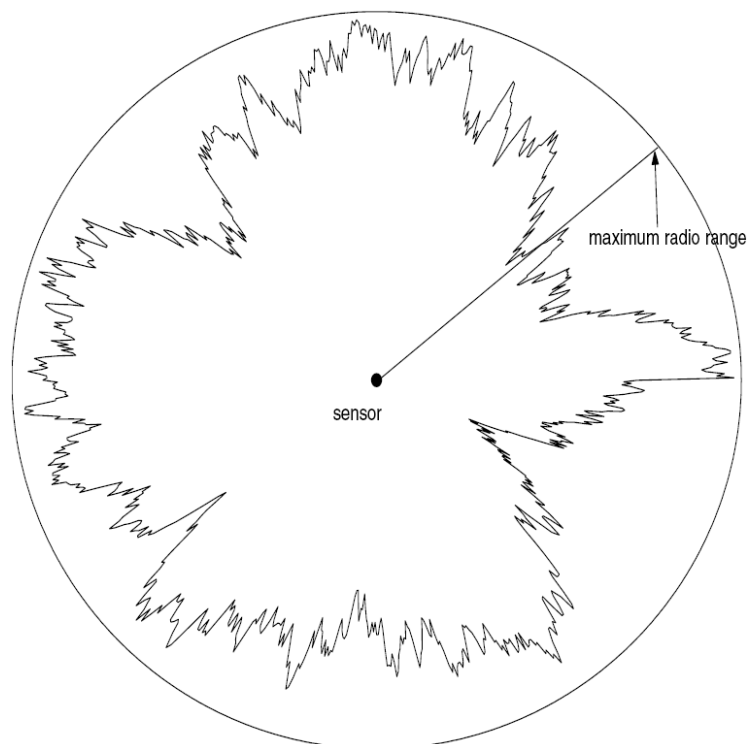
#### 3.1 MELHORIAS NA MEDIÇÃO DE ALCANCE

A distância estimada entre sensores é crucial para redução de erros quando se calcula a posição. O HECOPS utiliza heurísticas para abordar o problema e aumentar a precisão. A mais importante é introduzir uma compensação aos erros causados pela propagação de radio frequência. Além desta são propostas outras heurísticas.

##### 3.1.1 HEURÍSTICA

A propagação da rádio frequência de um sensor é representada como um círculo centrado no sensor. No mundo real, entretanto, o sinal de rádio de um sensor tem um padrão de propagação conforme mostrado na Figura 12. Isto significa que o alcance do

sinal tem uma propagação diferente em diferentes direções. Muitas condições ambientais afetam as ondas e causam diferenças nas medições de alcance. A intensidade de um sinal recebido em um sensor pode ser diferente do esperado [Ji 2004].

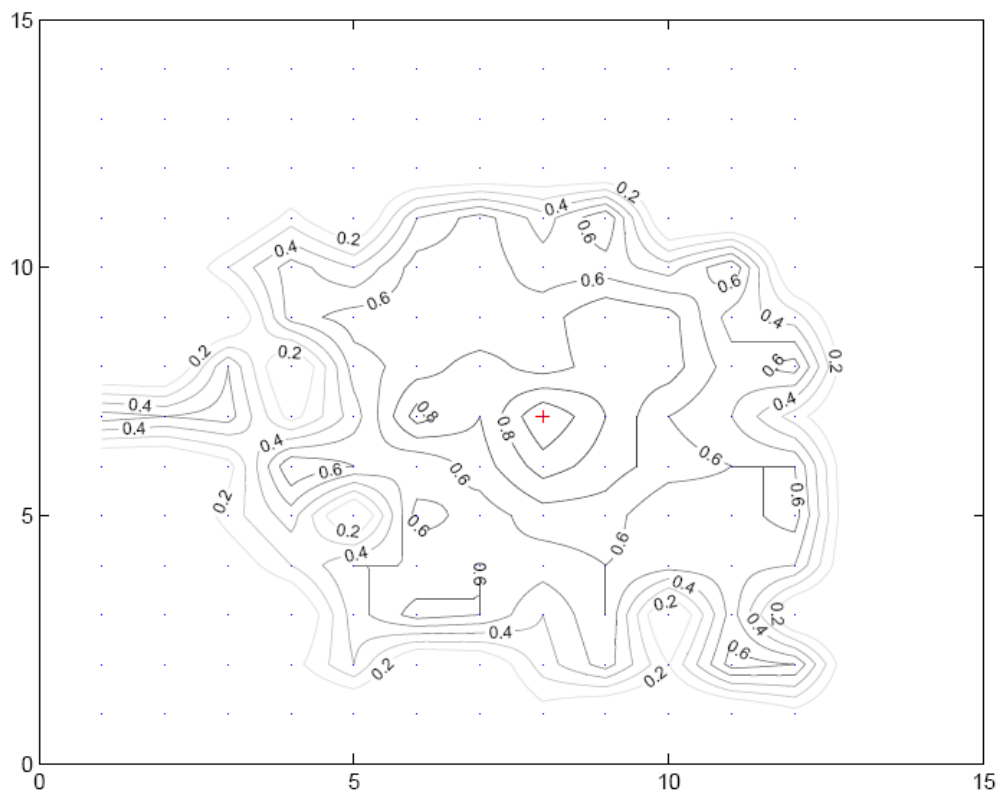


**Figura 12: Padrão de propagação irregular de RF em um sensor [Ji 2004].**

A taxa de recebimento de pacotes é diretamente proporcional à intensidade de sinal recebido (RSSI). Um estudo empírico indica que a distribuição da taxa de recebimento de pacotes considerada ao longo de uma distância é consideravelmente não uniforme como se pode ver na Figura 13 [Ganesan 2002]. Os contornos na figura exibem claramente a direção da propagação, sendo maior em algumas direções.

Mas a conclusão mais importante acerca desta figura, que é fundamental para o entendimento do problema em estudo, é de que, embora irregular, os níveis de RSSI podem ser vistos como círculos concêntricos. Isto significa que deve existir uma diferença entre a medição de alcance e a distância física. Esta distância será chamada aqui de “desvio”, e representa o erro causado por fatores de interferência como o da Figura 13.





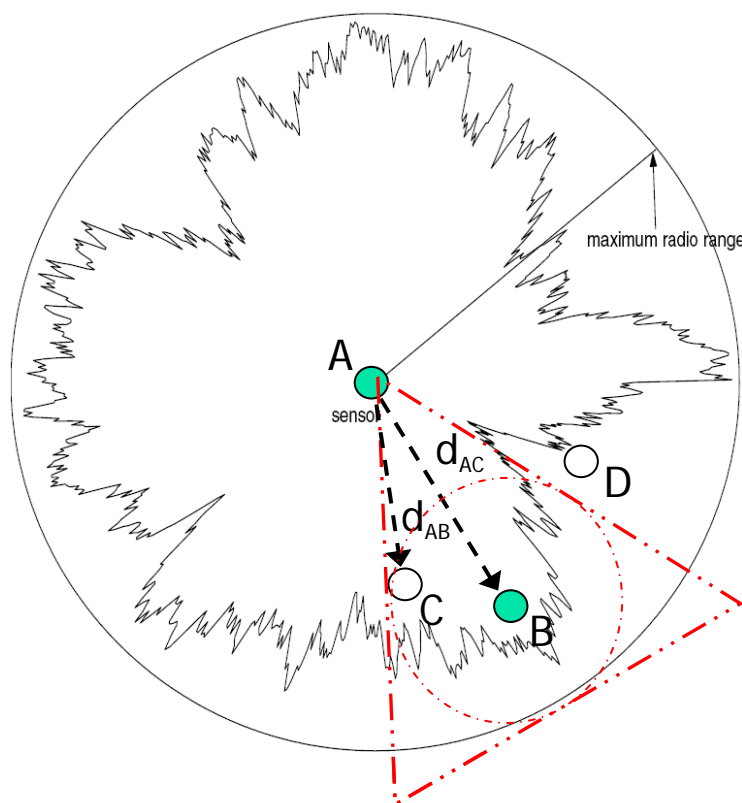
**Figura 13: Contorno das probabilidades de recepção de pacotes de um sensor central [Ganesan 2002].**

Considerando a propagação de rádio frequência em uma direção específica, a idéia é identificar eventuais desvios e informar aos sensores localizados ao longo desta direção. Estes sensores usam esta informação como fator de compensação no cálculo da posição individual. Este fator de compensação causa o efeito de calibragem no sistema. Em outras palavras o sistema pode calibrar as medições de alcance comparando em alguns casos a distância estimada pela medição com a distância real conhecida, ou mais próxima da real.

### 3.1.2 FUNÇÃO *tri*

A função *tri* é o elemento fundamental do algoritmo, pois através dela são identificados os desvios da propagação de RF explicados na seção anterior. Os desvios (*dev*) são detectados quando uma medida de alcance difere da distância calculada, como pode ser visto no exemplo da Figura 14. Os sensores *A* e *B* são âncoras e, portanto

conhecem a distância real entre eles. Caso a medida de alcance detectada em B não for igual à distância real, entende-se que existe um desvio na propagação de A para B.



**Figura 14: Exemplo do sensor C sendo afetado pelo desvio detectado AB; função  $tri$  (AB) medição de alcance AC calibrada pelo sensor B.**

A fórmula para o desvio ( $dev$ ) é a seguinte:

$$dev_{AB} = L_{real} / d_{AB} \quad (9)$$

Onde:

$dev_{AB}$  = desvio entre os sensores A e B ;

$L_{real}$  = distância entre os sensores A e B;

$d_{AB}$  = distância estimada AB, usando medição de alcance.

No mesmo exemplo o sensor C está localizado ao longo da direção do desvio  $dev_{AB}$ . Por intuição define-se aqui que C é afetado pelo mesmo valor de  $dev_{AB}$ . A lógica do algoritmo sugere que o sensor C deva ser informado do valor do desvio e assim poder usar no cálculo de sua posição. É interessante notar que para o sensor C, a referência  $d_{AC}$  embora venha de um sensor mais longe, é muito mais confiável que a

referência  $d_{BC}$ . O sensor  $D$  está fora da área da função  $tri$  e não é considerado, pois está muito longe da área afetada.

Para se definir a área onde a qual um sensor é afetado pelos desvios de seus vizinhos, foi criada uma função  $tri$  que determina uma área da função  $tri(AB)$ . O formato da área da função  $tri$  foi proposta baseado unicamente em deduções intuitivas. Este formato pode ser melhorado após análise dos resultados práticos. A área proposta inicialmente tem a forma de um triângulo como pode ser visto na Figura 14. Entretanto, devido ao grande esforço computacional para o cálculo de uma área triangular, a área afetada foi simplificada pela representação de um círculo centrado no sensor receptor. No exemplo da Figura 14, o sensor  $B$  é o centro de um círculo. O raio do círculo pode ser facilmente determinado pela razão da distância  $AB$  pela distância  $BC$ . Para simplificar ainda mais o esforço computacional, decidiu-se considerar a distância  $AB/2$  como o raio do círculo da área  $tri$ . Portanto  $C$  está na área se  $BC < AB/2$ . A área triangular original teria então a forma de um triângulo isósceles com altura igual 1,5 vezes a distância entre o par de sensores que detectam o desvio, isto é, um sensor emissor  $A$  e um receptor  $B$ . O nome  $tri$  vem do formato de triângulo. Um exemplo pode ser visto na Figura 14. Um sensor qualquer  $C$  uma vez localizado dentro de uma área  $tri(AB)$  passa a ter a possibilidade de ter sua medição de alcance  $AC$  é calibrada pelo sensor  $B$ . Um sensor  $C$  é dito ter a função  $tri$  se ele se encontrar dentro de qualquer área de função  $tri(AB)$  sendo que as confianças nas posições de  $A$  e  $B$  devem ser maiores que 0,80. A definição de nível confiança é explicada na seção 3.2

### 3.1.3 DESVIO PERMANENTE

Esta proposta é uma extensão da aplicação da função  $tri$ . No caso de um sensor detectar valores semelhantes de desvios ao seu redor, num raio de 360 graus ou menos. Pode-se supor então que a causa dos desvios possa ser na realidade, variações no equipamento de rádio transmissor ou receptor. A idéia então é considerar um valor  $dev$  permanente para compensar e calibrar este erro em todas as direções ou em um ângulo específico de transmissão. Uma parede cortando uma rede de sensores poderia ser detectada pelos diversos desvios e então estabelecer compensações que reduziriam as distorções. É importante notar que este tipo de informação não é recebida diretamente

dos transmissores e, portanto um protocolo adicional deve ser desenvolvido. Esta possibilidade não é explorada neste estudo.

### **3.1.4 TODOS OS SENSORES SÃO REFERÊNCIAS**

A propagação de rádio frequência é irregular em um ambiente anisotrópico. Considerando dois sensores *A* e *B*, provavelmente o valor da medição de alcance de *A* para *B* seria diferente do valor de *B* para *A*. O algoritmo é desenhado para considerar com duas referências independentes. A melhor referência, neste caso a mais confiável, será selecionada automaticamente pelo algoritmo e deve depender da topologia da rede e condições ambientais. O problema que pode surgir neste caso é o aumento da quantidade de referências e conseqüentemente aumento da complexidade de armazenar, transmitir e calcular. Neste estudo não foi testado este potencial, porém é muito claro que o algoritmo suportaria a implementação sem precisar maiores alterações.

### **3.1.5 TRANSMISSÃO EM DIFERENTES NÍVEIS DE POTÊNCIA**

Como demonstrado por Ganesan[Ganesan 2002] e em outro estudo por Lorincz [Lorincz 2004], na transmissão em vários níveis de potência, as ondas de rádio frequência se propagam de maneiras diferentes no meio e possibilitam resultados diferentes nas recepções . De acordo com Lorincz, variando-se a potência de transmissão se obtém diferentes medições e assim diversifica-se o conjunto de referências e de fato aumenta-se a precisão em algum metros na experiência realizada [Lorincz 2004].

O HECOPS pode aceitar as transmissões dos sensores em vários níveis de potência, sendo que cada potência será uma nova referência independente. Entretanto o aumento do número de referências implica em maior complexidade e conseqüentemente maior esforço computacional. Embora nenhum teste fosse realizado neste estudo, fica claro que o algoritmo não precisaria ser alterado para aceitar esta alternativa.

## **3.2 SISTEMA DE COORDENADAS**

Todos os sensores devem trocar informações e estabelecer por si mesmo um sistema de coordenadas, sem o suporte de uma central. A idéia básica é selecionar um

número de sensores chamados de referências (“*landmarks*”). Esta abordagem é similar ao de Wu e outros estudos, entretanto aqui todos os sensores podem servir como referências e também cada sensor seleciona seus próprios sensores de referência [Wu 2005].

Nesta seção é introduzido o modelo de distância euclidiana. Depois, um método para calcular a posição e selecionar as referências é apresentado. Lembrando que por simplicidade o algoritmo é demonstrado em duas dimensões, porém naturalmente pode ser implementado em três dimensões.

### 3.2.1 CÁLCULO DA POSIÇÃO

Um par de sensores pode estimar a distância que os separa através da medição de alcance. Existem várias técnicas para isso conforme explicado da seção 2.1. Entretanto, quando os sensores não estão adjacentes, a distância estimada se torna não trivial e requer estratégias de cálculo.

O cálculo sugerido para este problema de multi-lateração foi definido aqui como a função que minimiza o residual de erro, conforme demonstrado no capítulo anterior. O residual de erro se refere à diferença entre as distâncias medidas até as referências (MA) e das distâncias estimadas da posição calculada a cada iteração. O motivo da escolha deste método é o baixo esforço computacional [Savarese 2001, Niculescu, 2003].

Pode-se então considerar a seguinte função de minimização:

$$\text{Min } f(x, y) = \sum_{i=1}^4 \sqrt{(x_i - x)^2 + (y_i - y)^2} - d_i \quad (10)$$

Onde:

$i$  são as referências escolhidas;

$x$  e  $y$  são as coordenadas;

$d$  é a distância até as referências  $i$ .

Este método tem a vantagem de flexibilizar o esforço computacional do sensor na medida em que ele pode empregar mais ou menos tempo computacional, pela quantidade de iterações, conforme desejar mais ou menos precisão.

As estimativas iniciais podem ser obtidas pelo método Min-max [Savvides 2002] ou nos resultados calculados anteriormente. Quanto mais próximo do resultado real,

mais rápido o algoritmo converge. O algoritmo incrementa e decrementa as coordenadas  $x$  e  $y$  até o residual do resultado da função ou número de interações atingirem um valor pré-definido.

Teoricamente para o cálculo de posição, o número mínimo de referências necessárias é três, porém testes práticos mostraram que a probabilidade de três sensores estarem alinhados em uma mesma reta não pode ser ignorada. Neste caso a função teria dois resultados possíveis. Portanto o HECOPS procura usar quatro referências.

### 3.2.2 ESCOLHA DAS REFERÊNCIAS

Uma grande limitação de uma RSSF é que os sensores não usam rádios de alta potência de transmissão devido ao elevado consumo de energia, aumento de interferência na transmissão de dados e cobertura de alcance em sensores móveis [Niculesco 2001]. Isto resulta em um dispositivo mais simples e com capacidade reduzida de transmissão. Conseqüentemente nem todos os sensores se comunicam diretamente com os âncoras, necessitando estratégias de múltiplos-saltos (*multi-hop*). Este problema é chamado problema de âncoras esparsas. Ele é atacado aqui, considerando-se não somente as âncoras para o cálculo, mas qualquer sensor da rede. Os sensores escolhidos para o cálculo são chamados neste estudo de “referências”. Um sensor é considerado uma referência para outro sensor se puder ser estabelecida uma medição de alcance e se for possível a transmissão das coordenadas de posição.

Esta solução é semelhante aos algoritmos recursivos como proposto por [Albowicz 2001, Savvides 2001, e Wu 2004]. Essas soluções, que se utilizam de algoritmos distribuídos, trabalham por meio de múltiplos saltos para difundir as informações de localização entre os sensores da rede. Nessa abordagem, um sensor estima a sua posição com base nas informações de posição de pelo menos três outros sensores que já conheçam suas posições. Uma vez estimada a sua posição, este sensor divulga a sua posição aos seus vizinhos para auxiliá-los na descoberta de suas próprias posições. Desta forma, os sensores que acabaram de calcular suas posições auxiliam outros sensores a calcularem as suas, até que todos os sensores da rede possuam uma posição estimada.

A proposição difere das outras abordagens quanto ao número de referências usadas para o cálculo de posição. Praticamente todas as outras soluções procuram usar o

máximo de referências para tentar reduzir o número de erros [Wu 2004]. Nesta abordagem aqui proposta, ao contrário, são utilizadas um número de quatro referências no cálculo. À medida que referências com baixa confiabilidade são desconsideradas no cálculo, evita-se a propagação de erros.

Esta abordagem aqui proposta somente é possível devido ao estabelecimento de um sistema de ranqueamento da confiabilidade na posição do sensor. Quanto mais próxima do real for a estimativa da posição, maior a confiança atribuída a um sensor.

### 3.2.3 ATRIBUINDO VALORES DE CONFIANÇA

Um valor é atribuído a cada sensor com relação à confiança na sua própria posição. Assim é possível estabelecer um sistema de ranqueamento dos sensores de toda a rede.

Como se usa um sistema de ranqueamento semelhante ao do algoritmo Hop-Terrain, decidiu-se adotar para o HECOPS o mesmo termo “confiança” para representar o valor deste nível. Também foi mantido o limite de confiança entre 0 e 1 para cada sensor [Savarese 2002].

Um sensor âncora qualquer, do qual sempre se conhece a posição *a priori*, tem por definição a confiança máxima e valor igual a 1. Aos demais sensores devem ser atribuídos valores menores de confiança. Por dedução definiu-se que, para a seleção de melhores referências, os âncoras deveriam ter importância muito maior que os demais, visto que conhecem realmente suas posições. Portanto para que o nível de confiança de um sensor não âncora nunca atingisse o valor 1, e nem valores próximos ao valor 1, limitou-se este a um valor máximo mantendo uma faixa de intervalo. A escolha do valor deste parâmetro foi realizada simultaneamente a escolha dos parâmetros de cálculo que definem a proporção de importância entre referências e respectivas funções *tri*.

Inicialmente foi definido intuitivamente que para o cálculo de confiança de um sensor, a média das confianças dos sensores selecionados como referências teriam um peso proporcional de 70% deste valor. A média das confianças dos sensores selecionados eventualmente na função *tri* representariam os outros 30% deste valor. Caso não existisse função *tri* para um determinado sensor, esta parcela de 30% seria zero. Estes parâmetros definidos em 0,7 e 0,3 asseguram que o valor calculado de *C* dependa mais da confiança dos sensores vizinhos, como se espera.

Fixando-se temporariamente os parâmetros em 0,7 e 0,3, seguiu-se um teste para se definir o parâmetro do valor mínimo necessário de confiança de um sensor para participar de uma função *tri*. Valores baixos de confiança podem aumentar o erro no cálculo de posição, ao invés de reduzir. O teste consistiu de uma simulação em um cenário com uma população de sensores com 10% de âncoras e 10% de erros acrescidos a cada medição e média de 24,6 sensores vizinhos conectados. Uma melhor explicação a respeito do simulador e seus parâmetros é feita na seção 4.1. Neste teste variou-se o limite mínimo da confiança necessário para um sensor participar de uma função *tri*. Como resultado observou-se da média de erro das posições O resultado pode ser visto na Figura 15.

limite mínimo fixado para confiança de um sensor para participar de uma função <i>tri</i>	resultado da média de erro das posições
95	2,33
90	2,33
85	2,33
80	2,11
75	6,01
70	16,28
65	15,46

**Figura 15: Estudo do limite mínimo fixado para confiança de um sensor para participar de uma função. Escolhido valor 80 ou 0,8 .**

Observando os resultados da figura, escolheu-se o valor de 80 ou 0,8 porque resultou em menor valor de média de erros das posições. Espera-se assim, que esta faixa de intervalo, entre 1 (valor do âncora) e 0,8 (valor máximo do não âncora), reflita em um impacto maior nos cálculos.

Com a definição deste parâmetro foi realizado um novo teste para ajustar os parâmetros de proporcionalidade 0,7 e 0,3. Os parâmetros da simulação foram os mesmos do teste anterior e resultaram na Figura 16. Observando os resultados da Figura 16, escolheu-se os valores de 0,75 e 0,25 porque para 75% de peso resultou em menor valor de média de erros das posições.

Todos os parâmetros propostos até agora deveriam ser revistos no futuro com o objetivo de minimizar os erros. Entretanto serão mantidos assim nesta dissertação, pois o estudo será concentrado na variação de outros parâmetros a serem apresentados no próximo capítulo.



peso (%) atribuído às referências no cálculo de C	resultado da média de erro das posições
85	45,10
80	43,56
75	40,01
70	60,04
65	59,63
60	56,89

**Figura 16: Estudo do peso (%) atribuído às referências no cálculo de C. Escolhido valor 75%.**

Considerando então um sensor  $n$  conectado a seus sensores vizinhos, fica definido aqui que a sua confiança ( $C$ ) é calculada pela seguinte fórmula:

$$C_n = \frac{\sum_{i=1}^3 C_i \times 0,75 + C_{tri(i,n)} \times 0,25}{3} \times 0,8 \quad (11)$$

Onde

$C_n$  = confiança no sensor  $n$  ;

$i$  = referência selecionada; seleciona-se os 3 maiores resultados da fórmula ( $C_i \times 0,75 + C_{tri_{in}} \times 0,25$ ) aplicado aos sensores conectados;

$C_i$  = confiança no sensor  $i$  ;

$C_{tri(i,n)}$  = valor da confiança do sensor que calibra uma função  $tri$ , valor igual a zero se menor que 0,8 ou se não existir função  $tri$ .

Um sensor não âncora nunca atinge  $C=1$ , não importa quão confiável sejam as informações disponíveis. Isto é devido ao fato de somente o sensor âncora saber sua posição. Os outros sensores não âncora sempre terão valores estimados.

A medição de alcance até uma âncora é uma informação confiável. Porém a precisão/confiabilidade pode ser aumentada se esta medição for calibrada pelo desvio ( $dev$ ) explicado na seção 3.2.1 . Portanto o fator  $tri$  na fórmula (11) pode tornar um sensor mais confiável e provável de ser selecionado como referência.

Somente sensores com posições confiáveis podem ser escolhidos para serem referência, caso contrário aumentar-se-ia a incerteza da posição. Por isso somente valores de  $tri$  maiores que 0,8 são considerados com a fórmula (11).

As constantes 0,75 e 0,25 asseguram que o valor calculado de  $C$  dependa mais da confiança dos sensores vizinhos, como se espera. A função  $tri$  ajuda a incrementar o  $C$ ,

entretanto por si só não pode elevar o valor final de  $C$  além de um limite. Caso contrário um sensor âncora poderia ser preterido na escolha como referência.

### 3.3 FUNCIONAMENTO

Nesta seção é explicado como os sensores armazenam e trocam informações. Cada sensor armazena em uma tabela  $t$  todas as informações referentes aos sensores referências que estiverem na sua vizinhança. Cada linha da tabela representa uma referência. A tabela é transmitida linha por linha para os sensores vizinhos. Baseado nas informações da tabela que está sendo recebida e informações de sua própria tabela, cada sensor receptor tenta identificar funções  $tri$ . Quando, ao receber uma nova linha, for identificada uma função  $tri$ , esta informação passa a ser então armazenada na respectiva linha da tabela do sensor receptor.

#### 3.3.1 TABELA DE DADOS

Cada sensor da rede mantém uma tabela  $t$  com as informações básicas para o cálculo de posição e identificação da função  $tri$ . A primeira linha é reservada para os parâmetros do próprio sensor que são: ID do sensor receptor ( $ID_n$ ), coordenada  $x$  da sua posição ( $X_n$ ), coordenada  $y$  da sua posição ( $Y_n$ ), desvio permanente estimado ( $D_n$ ), valor da confiança de sua posição ( $C_n$ ), número máximo de linhas da tabela ( $L_n$ ), e valor na menor confiança armazenada na tabela ( $C_{min}$ ).

Todas as outras linhas da tabela armazenam informações das referências como visto na Figura 17. Cada linha contém informações respectivas a uma referência que são: ID do sensor transmissor ( $ID_1$ ), ID do sensor receptor ( $ID_2$ ), coordenada  $x$  da sua posição do transmissor ( $X_1$ ), coordenada  $y$  da sua posição do transmissor ( $Y_1$ ), valor da medição de alcance (neste caso valor da intensidade do sinal) ( $S_{1,2}$ ), desvio estimado ( $D_{1,2}$ ), valor da confiança do  $ID_1$  ( $C_1$ ), ID do sensor que formou a função  $tri$  ( $ID_{tri}$ ), e valor da confiança de  $ID_1$  ( $C_{tri}$ ).

ID <sub>1</sub>	ID <sub>2</sub>	X <sub>1</sub>	Y <sub>1</sub>	S <sub>12</sub>	D <sub>12</sub>	C <sub>1</sub>	ID <sub>tri</sub>	C <sub>tri</sub>
		200	300			0.73		
b	a	30	400	197	1.02	1	c	1
c	a	150	220	94		1		
d	a	50	50	292		0.51		
e	a	80	20	305	1.01	0.75	c	1

**Figura 17: Exemplo de tabela de dados de um sensor A que armazena referências de quatro outros sensores B, C, D e E.**

A quantidade máxima de linhas armazenadas no sensor é determinada pela capacidade de *hardware*, taxa de transmissão e máximo erro desejado. O valor máximo de linhas sugerido é de 10 e será demonstrado mais adiante.

Todos os sensores iniciam com suas tabelas vazias, exceto os âncoras, que já iniciam com uma linha referente à sua própria posição conhecida. A cada linha recebida, cada sensor decide se a inclui na tabela ou não. Caso a referência já exista na sua tabela a linha pode ser atualizada neste momento. Ao tentar incluir uma linha e a tabela estiver com o limite máximo, então a linha correspondente ao menor valor de  $C$  é substituída pela referência que está sendo recebida, desde que a linha recebida tenha um valor de  $C$  superior ao menor valor de  $C$  da tabela.

### 3.3.2 ATUALIZAÇÃO DA TABELA DE DADOS

É assumido que deva existir uma seqüência definida de transmissão dos sensores. Nos exemplos deste estudo é considerado a seqüência ABCD. Os sensores transmitem suas tabelas linha por linha. Cada sensor receptor (ID<sub>2n</sub>) atualiza sua tabela quando um sensor transmissor (ID<sub>1n</sub>) envia informações. A atualização usa o seguinte algoritmo conforme o pseudocódigo a seguir considerando para todo o receptor ID<sub>2n</sub> :

**lê** pacote e define ID<sub>1n</sub> , C<sub>1n</sub> e valor S<sub>12n</sub>

cria linha na tabela e armazena sensor ID<sub>1n</sub> e S<sub>12n</sub>

**se** a tabela cheia      **então** substitui linha com menor valor de C<sub>1</sub>

**senão** inclui linha na tabela

**repete** ler linha até última linha

verifica se existe função *tri* (transmissor, receptor, sensor da linha n)

**se** verdadeiro e C<sub>1</sub> > C<sub>1min</sub>      **então** inclui/substitui linha na tabela

**senão** lê próxima linha

seleciona máximo 4 linhas com maiores  $C_1$

**se** somente 1 referência **então** assume coordenada  $(X_{In}, Y_{In} + d_{AB})$  do transmissor

**senão** calcula coordenada  $(X_{In}, Y_{In})$

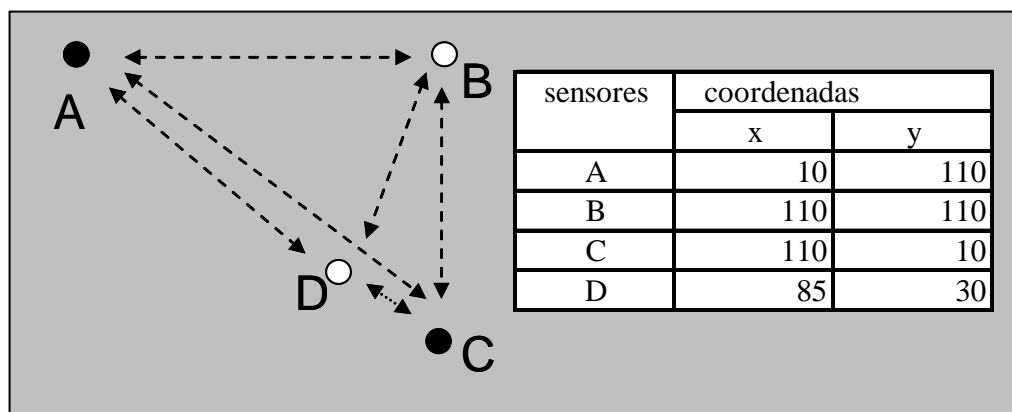
calcula  $C_1$  de sua posição.

Um sensor precisaria ler apenas alguns pacotes transmitidos para determinar a medição de alcance  $S_{AB}$ , porém acaba lendo todas as linhas transmitidas procurando identificar uma função *tri*. Quando uma linha estiver sendo recebida e indicar a entrada de um novo sensor que ainda não estava na tabela, então uma nova linha deve ser incluída. Entretanto o número de linhas não pode exceder um determinado limite estipulado. Quando o número de linhas atingir o limite estabelecido, a nova linha deve substituir a linha da tabela com menor valor de confiança. Isto assegura que a tabela mantenha sempre somente linhas referentes às melhores referências. O problema em não se manter todas as linhas possíveis é a diminuição da probabilidade de encontrar funções *tri*.

Caso somente uma referência esteja disponível para o cálculo da posição, o sensor assume a posição  $(X, Y + d_{AB})$  do transmissor. Esta primeira estimativa de posição é pobre, porém melhora sua precisão com o decorrer das iterações.

### 3.3.3 EXEMPLO

Para explicar melhor o funcionamento do algoritmo é apresentado a seguir um exemplo de trocas de informações nos sensores do exemplo da Figura 18. Neste exemplo *A* e *C* são âncoras e *B* e *D* são sensores que tentam estimar suas posições.



**Figura 18: Exemplo de funcionamento de rede com dois sensores âncoras A e B e dois sensores com posição desconhecida B e D, todos se comunicam.**

Inicialmente as tabelas de todos os sensores estão vazias, exceto as tabelas dos âncoras como visto na Figura 19.

tabela do sensor A								
ID <sub>1</sub>	ID <sub>2</sub>	X <sub>1</sub>	Y <sub>1</sub>	S <sub>12</sub>	D <sub>12</sub>	C <sub>1</sub>	ID <sub>tri</sub>	C <sub>tri</sub>
		10	110			1,00		

tabela do sensor B								
ID <sub>1</sub>	ID <sub>2</sub>	X <sub>1</sub>	Y <sub>1</sub>	S <sub>12</sub>	D <sub>12</sub>	C <sub>1</sub>	ID <sub>tri</sub>	C <sub>tri</sub>

tabela do sensor C								
ID <sub>1</sub>	ID <sub>2</sub>	X <sub>1</sub>	Y <sub>1</sub>	S <sub>12</sub>	D <sub>12</sub>	C <sub>1</sub>	ID <sub>tri</sub>	C <sub>tri</sub>
		110	10			1,00		

tabela do sensor D								
ID <sub>1</sub>	ID <sub>2</sub>	X <sub>1</sub>	Y <sub>1</sub>	S <sub>12</sub>	D <sub>12</sub>	C <sub>1</sub>	ID <sub>tri</sub>	C <sub>tri</sub>

**Figura 19: Exemplo de funcionamento – estado inicial, tabelas vazias exceto as dos âncoras.**

Após a primeira transmissão do sensor A as tabelas estão atualizadas conforme Figura 20. O sensor B recebe e armazena sua primeira referência que é a medição de alcance (102) e as coordenadas de A (10, 110). Ele então estima sua posição (10, 212) e calcula sua confiança (0,20). O sensor C armazena igualmente a linha de referência relativa ao sensor A, porém como é uma âncora, não precisa calcular posição nem confiança.

tabela do sensor A								
ID <sub>1</sub>	ID <sub>2</sub>	X <sub>1</sub>	Y <sub>1</sub>	S <sub>12</sub>	D <sub>12</sub>	C <sub>1</sub>	ID <sub>tri</sub>	C <sub>tri</sub>
		10	110			1,00		

tabela do sensor B								
ID <sub>1</sub>	ID <sub>2</sub>	X <sub>1</sub>	Y <sub>1</sub>	S <sub>12</sub>	D <sub>12</sub>	C <sub>1</sub>	ID <sub>tri</sub>	C <sub>tri</sub>
		10	212			0,20		
A	B	10	110	102		1,00		

tabela do sensor C								
ID <sub>1</sub>	ID <sub>2</sub>	X <sub>1</sub>	Y <sub>1</sub>	S <sub>12</sub>	D <sub>12</sub>	C <sub>1</sub>	ID <sub>tri</sub>	C <sub>tri</sub>
		110	10			1,00		
A	C	10	110	146		1,00		

tabela do sensor D								
ID <sub>1</sub>	ID <sub>2</sub>	X <sub>1</sub>	Y <sub>1</sub>	S <sub>12</sub>	D <sub>12</sub>	C <sub>1</sub>	ID <sub>tri</sub>	C <sub>tri</sub>
		10	225			0,20		
A	D	10	110	115		1,00		

**Figura 20: Exemplo de funcionamento – após transmissão do sensor A, é adicionada uma linha em cada tabela dos outros sensores.**

Para fins de explicação, é importante notar que o sensor *C* também conseguiria identificar o primeiro desvio de 3% entre *A* e *C*, pois sua medição de alcance informa 146, e a distância euclidiana  $d_{AC}$  calculada indica valor de 141. Esta informação de desvio poderia ser armazenada na coluna  $D_{12}$ , porém ainda não se faz, pois não se sabe se poderá ser útil no futuro. Também é importante lembrar que este cálculo somente é relevante porque a confiança de cada sensor é alta. O sensor *D* procede semelhantemente a *B*, incluindo uma linha e calculando posição e confiança.

Após a primeira transmissão do sensor *B*, as tabelas estão atualizadas conforme Figura 21. Todos os sensores armazenam uma linha referente às coordenadas estimadas de *B* (10, 212) e sua confiança (0,2). O sensor *D* vai além e calcula uma nova posição (10, 225) e uma nova confiança, mas ainda desconsiderando *B* por ter baixa confiança.

tabela do sensor A									
ID <sub>1</sub>	ID <sub>2</sub>	X <sub>1</sub>	Y <sub>1</sub>	S <sub>12</sub>	D <sub>12</sub>	C <sub>1</sub>	ID <sub>tri</sub>	C <sub>tri</sub>	
		10	326			1,00			
B	A	10	212	102		0,20			

tabela do sensor B									
ID <sub>1</sub>	ID <sub>2</sub>	X <sub>1</sub>	Y <sub>1</sub>	S <sub>12</sub>	D <sub>12</sub>	C <sub>1</sub>	ID <sub>tri</sub>	C <sub>tri</sub>	
		10	212			0,20			
A	B	10	110	103		1,00			

tabela do sensor C									
ID <sub>1</sub>	ID <sub>2</sub>	X <sub>1</sub>	Y <sub>1</sub>	S <sub>12</sub>	D <sub>12</sub>	C <sub>1</sub>	ID <sub>tri</sub>	C <sub>tri</sub>	
		110	10			1,00			
A	C	10	110	144		1,00			
B	C	10	212	104		0,20			

tabela do sensor D									
ID <sub>1</sub>	ID <sub>2</sub>	X <sub>1</sub>	Y <sub>1</sub>	S <sub>12</sub>	D <sub>12</sub>	C <sub>1</sub>	ID <sub>tri</sub>	C <sub>tri</sub>	
		10	225			0,24			
A	D	10	110	115		1,00			
B	D	10	212	85		0,20			

**Figura 21: Exemplo de funcionamento – após transmissão do sensor B, sensor D agora tem duas referências e recalcula posição e confiança.**

O processo continua com a transmissão em seqüência dos sensores C e D, completando um ciclo de transmissão, ou um processo iterativo de cálculo.

Considerando-se agora o segundo ciclo de transmissão, exatamente no instante após o sensor B transmitir e antes da transmissão de C, como mostrado na Figura 21.

tabela do sensor C									
ID <sub>1</sub>	ID <sub>2</sub>	X <sub>1</sub>	Y <sub>1</sub>	S <sub>12</sub>	D <sub>12</sub>	C <sub>1</sub>	ID <sub>tri</sub>	C <sub>tri</sub>	
		110	10			1,00			
A	C	10	110	146		1,00			
B	C	59	61	104		0,40			
D	C	101	20	34		0,40			

tabela do sensor D									
ID <sub>1</sub>	ID <sub>2</sub>	X <sub>1</sub>	Y <sub>1</sub>	S <sub>12</sub>	D <sub>12</sub>	C <sub>1</sub>	ID <sub>tri</sub>	C <sub>tri</sub>	
		101	20			0,40			
A	D	10	110	115		1,00			
B	D	59	61	85		0,20			
C	D	110	10	34		1,00			

**Figura 22: Exemplo de funcionamento – antes da transmissão do sensor C, segundo ciclo.**

Analisando-se agora a partir do início da transmissão do sensor *C*:

O sensor *D* ao receber a transmissão de *C*, primeiramente atualiza sua medição de alcance (34), e posição de *C* (101, 20). Ambas as informações não sofreram alterações do primeiro ciclo, mas eventualmente poderiam. O sensor *D* passa então a ler as linhas que estão na tabela de *C*. A primeira linha lida informa que o sensor transmissor recebeu também do sensor *A*. O sensor *D* procura então em sua tabela, especificamente na coluna  $ID_1$ , se também recebeu dados deste mesmo sensor. Caso não encontrar significa que não está conectado a ele e passa então a ler a próxima linha. Caso encontre na tabela indica que pode haver uma função *tri*. Ou seja, *C* e *D* estão ambos recebendo sinal de um terceiro sensor *A* e estão suficientemente próximos para serem influenciados pelo mesmo sensor transmissor.

Para saber se estão suficientemente próximos, comparou-se neste exemplo as distâncias euclidianas  $d_{CD}$  e  $d_{AD}$ . A condição para haver o *tri* considerada aqui é que  $d_{CD} < (d_{AD} / 2)$ . Isto assegura que *D*, que está calculando sua posição, deve estar suficientemente perto de *C* para ser calibrado pelo sensor *A*. Outra condição a ser atendida é de que os sensores da função *tri* tenham confiança igual ou maior que 0,8. Caso a confiança da linha encontrada não atinja este mínimo, o sensor passa a ler uma nova linha. Foi o que aconteceu neste caso, onde se formou então a função *tri(AC)* conforme mostra a Figura 23. A tabela de *D* é então atualizada na linha correspondente ao sensor *A*. Armazena o ID do sensor que calibra a medição que é o sensor *C*. Também armazena a confiança de *C*, e o desvio *dev AC* de 3% (103).

Resumindo em outras palavras: o sensor *C* identificou um desvio vindo de *A*, e como está relativamente próximo do sensor *D*, este mesmo valor de desvio também é assumido como válido para o sensor *D*.



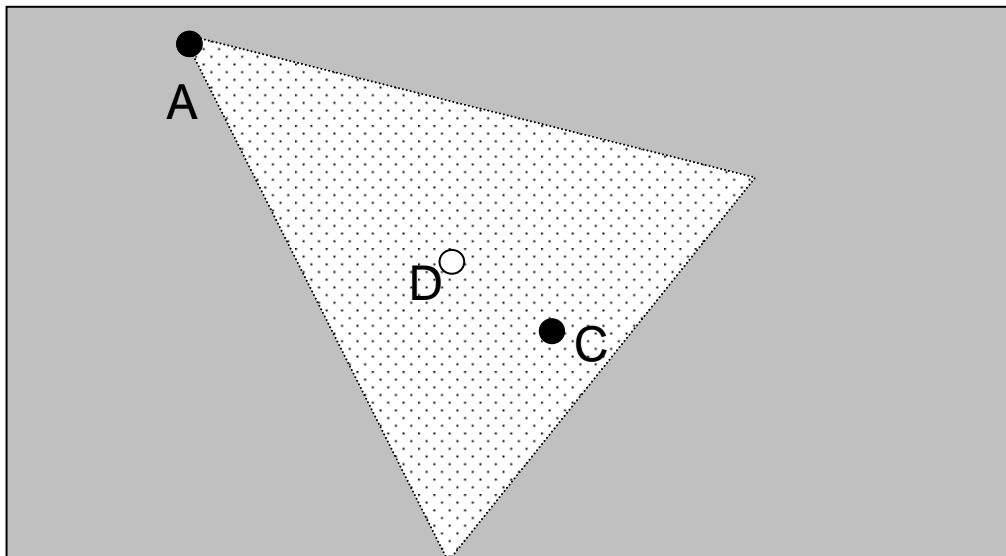


Figura 23: Exemplo de funcionamento – sensor *D* identifica a função *tri* (*AC*), a área hachurada indica a área afetada pelo desvio entre *A* e *C*.

tabela do sensor C								
ID <sub>1</sub>	ID <sub>2</sub>	X <sub>1</sub>	Y <sub>1</sub>	S <sub>12</sub>	D <sub>12</sub>	C <sub>1</sub>	ID <sub>tri</sub>	C <sub>tri</sub>
		110	10			1,00		
A	C	10	110	146	102	1,00		
B	C	59	61	104		0,40		
D	C	101	20	34		0,40		

tabela do sensor D								
ID <sub>1</sub>	ID <sub>2</sub>	X <sub>1</sub>	Y <sub>1</sub>	S <sub>12</sub>	D <sub>12</sub>	C <sub>1</sub>	ID <sub>tri</sub>	C <sub>tri</sub>
		99	21			0,47		
A	D	10	110	115	102	1,00	C	1,00
B	D	59	61	85		0,40		
C	D	110	10	34		1,00		

Figura 24: Exemplo de funcionamento – após transmissão do sensor *C*, segundo ciclo, o sensor *D* identifica a função *tri* (*AC*) e armazena na linha *AD* da tabela.

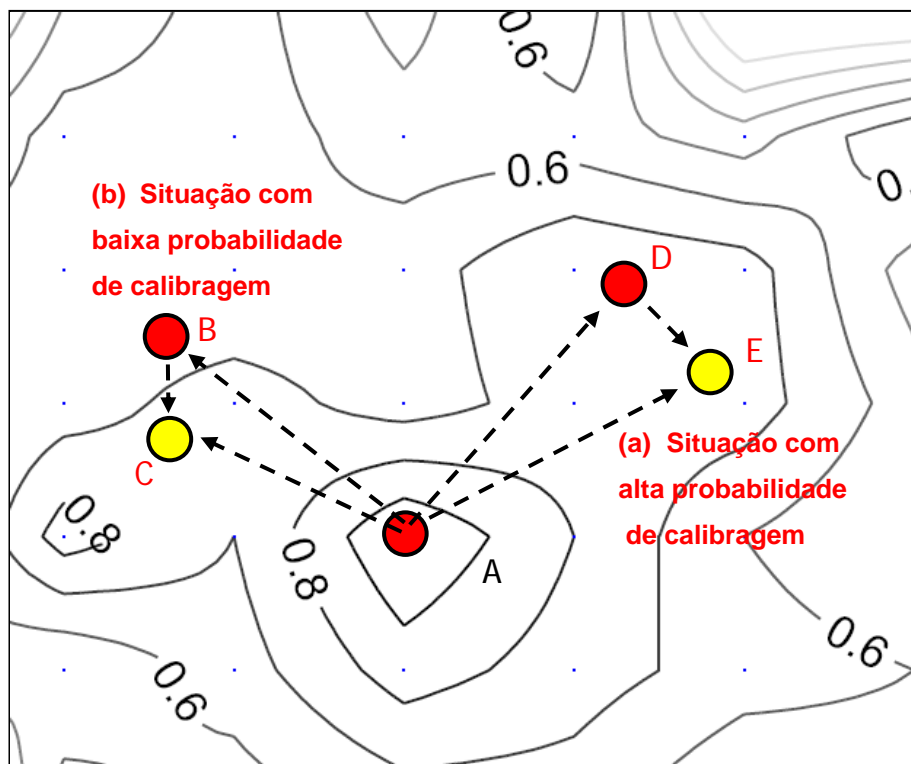
## 4 RESULTADOS EXPERIMENTAIS

O desempenho do algoritmo foi avaliado através de simulações considerando-se várias topologias, conectividade e erros de medição de alcance. O desempenho foi confrontado com o desempenho do algoritmo Hop-Terrain. Para tal, foi escrito um programa para o ambiente do *software* Matlab [Matlab 2006] para simular o funcionamento, analisar e visualizar o comportamento frente ao algoritmo atual.

### 4.1 SIMULADOR

Todos os parâmetros escolhidos para o simulador deste trabalho foram baseados no estudo de um algoritmo similar, o Hop-Terrain [Savarese 2002]. Apenas reduziu-se o número de cenários e sensores devido ao elevado tempo computacional.

Todos os pontos dos gráficos a seguir representam as médias de 20 cenários de uma rede com 100 sensores. Os sensores são posicionados randomicamente com uma distribuição normal, em uma área quadrada. A população de âncoras é definida randomicamente dentre os 100 sensores. O erro adicionado à medição de alcance (MA) é considerado da seguinte forma: para todo o sensor que estiver transmitindo, cada sensor que estiver recebendo este sinal têm adicionados um valor randômico, tendo o valor real como mínimo. Isto tenta representar a borda irregular (serrilhada) do círculo de propagação de RF. O valor destes erros adicionados é definido pela distribuição normal de um valor até um máximo especificado. Para tentar simular a propagação irregular da RF, em que as ondas não são círculos perfeitos, como demonstrado na Figura 25, considerou-se o sensor dividido em dois hemisférios. Todos os sensores receptores que se encontram em um hemisfério do sensor emissor têm ainda somado um valor de erro de 10%. Desta forma as medições recebidas de um lado do sensor diferem sempre dos valores recebidos do outro lado. O alcance da transmissão dos sensores é o parâmetro que define a quantidade média de sensores conectados, ou aqui chamado de conectividade. Quanto maior o alcance mais sensores são encontrados. Um nível de conectividade 5 significa uma média de 5 sensores conectados a cada sensor.



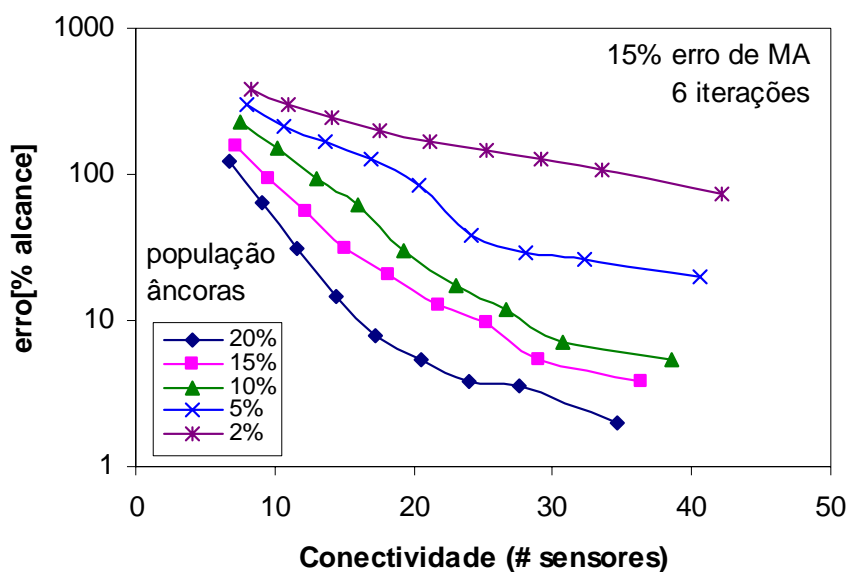
**Figura 25: Exemplo de situações da função *tri*: (a) baixa probabilidade de calibragem e (b) alta probabilidade de calibragem.**

Para permitir a comparação entre diferentes cenários, os erros de medição, assim como os erros de posição são normalizados pelo alcance da transmissão. Isto quer dizer que um erro de 100% na posição significa um erro igual ao do alcance de transmissão.

## 4.2 RESULTADOS

O desempenho geral do algoritmo pode ser exemplificado na Figura 26 em função da conectividade e população de âncoras. Foram simulados cenários com erros de alcance máximos nos valores de 20%, 15%, 10%, 5% e 1%. Como as curvas dos gráficos resultaram semelhantes, optou-se por apresentar somente os gráficos referentes aos erros de 15% simplesmente por parecerem de melhor visualização.

Como pode ser observado na Figura 26, considerando uma população de 15% de âncoras, as posições estimadas têm uma média de erros abaixo de 100% em cenários somente com nível de conectividade 10 ou mais. Em situações extremas, onde a população de âncoras é muito baixa (2%), os erros são menores que 100% somente com uma conectividade de ao menos 35.



**Figura 26: Desempenho do HECOPS - médias dos erros de posição variando-se a conectividade.**

A Figura 27 é similar à Figura 26, porém agora os resultados são confrontados com o algoritmo Hop-Terrain. Para um nível de conectividade entre 10 e 28. O HECOPS sempre consegue 10% a menos de erros que o Hop-Terrain. O HECOPS precisa de nível 15 de conectividade para atingir 30% de erros. O Hop-Terrain precisa de um nível 22, quase 50% mais. Devido a este fato de precisar de um menor número de referências comparado com o Hop-Terrain, o HECOPS pode ser considerado mais robusto em relação ao problema de âncoras esparsas (*sparse anchor problem*). A estratégia de descartar as piores referências, mesmo quando não existem muitas disponíveis, apresenta melhor resultado.

Em cada iteração todos os sensores conectados conseguem mais informações das referências e assim podem melhorar suas estimativas de posição. O HECOPS consegue uma precisão boa após duas iterações como pode ser visto na Figura 28. O Hop-Terrain chama esta fase de refinamento. Pelos resultados o Hop-terrain inicialmente tem uma estimativa melhor, mas estabiliza em um nível de erro superior.

Embora o HECOPS use apenas quatro referências, cada sensor armazena muitas outras na sua tabela. Como já explicado, o objetivo é aumentar a probabilidade de detecção de função *tri*. O problema em se usar uma tabela grande é aumentar a comunicação de dados desnecessários. Outro problema é a capacidade limitada de memória do dispositivo.

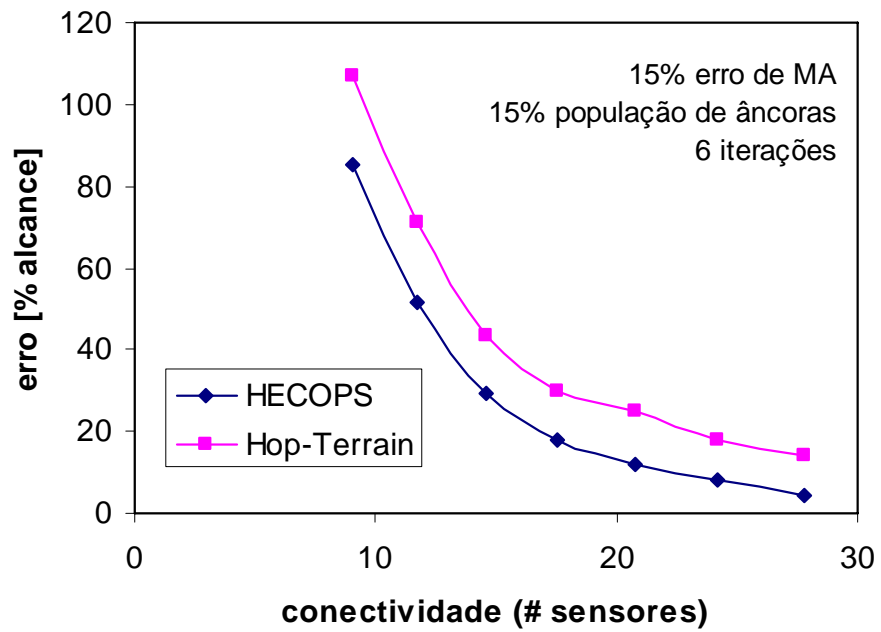


Figura 27: Desempenho do HECOPS - comparação da média de erros de posição: HECOPS e Hop-Terrain.

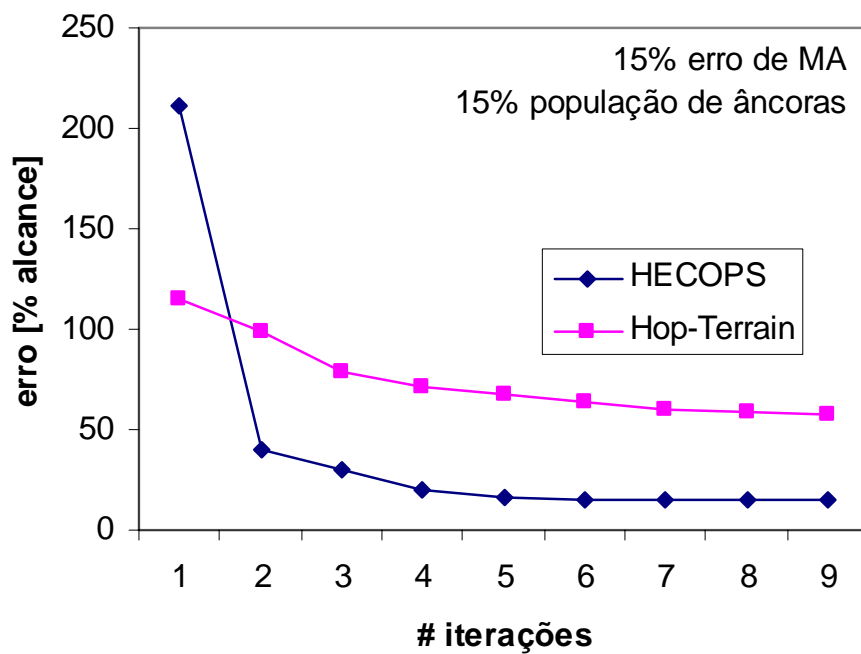
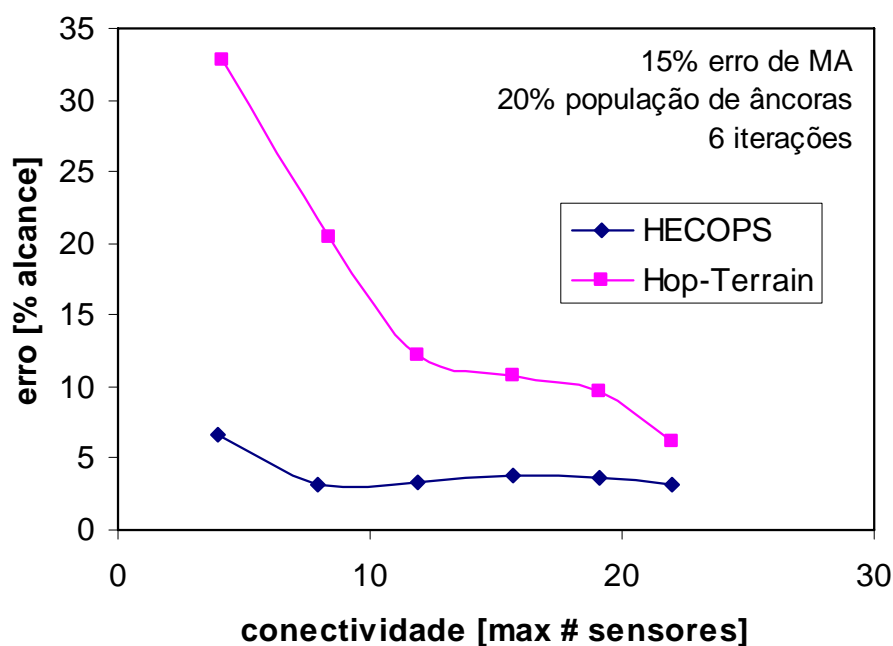


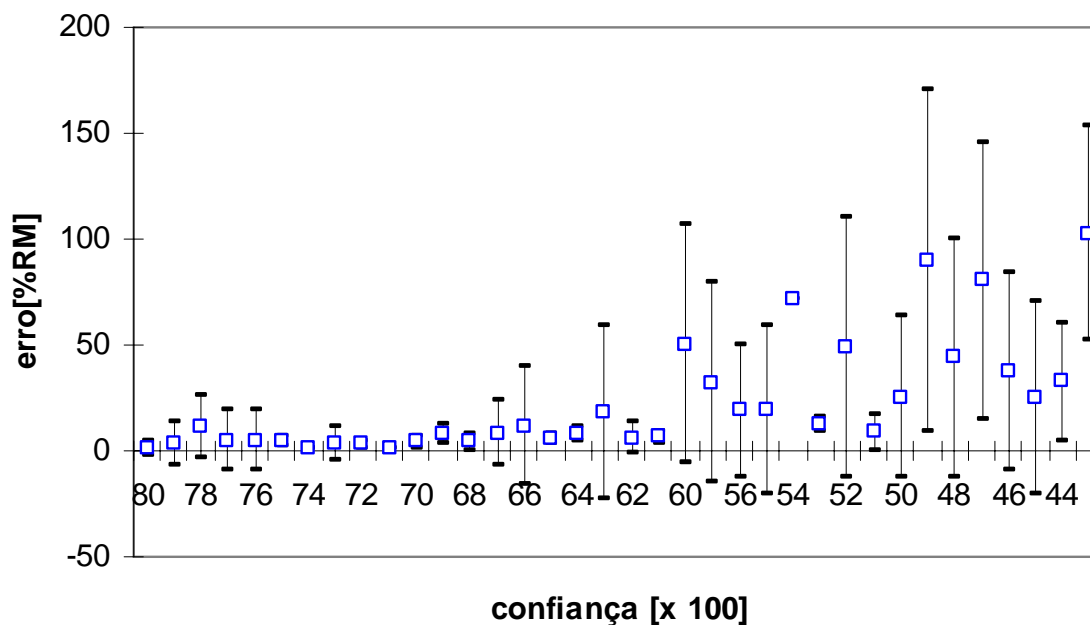
Figura 28: Desempenho do HECOPS - comparação de resultados ao longo de iterações (HECOPS e Hop-Terrain).

Na Figura 29 foram limitadas o número máximo de referências em cada tabela. A conclusão é de que para o HECOPS não ocorreu uma perda significativa de precisão. Para o Hop-Terrain, entretanto, os erros caíram abaixo de 10% somente após o limite de 22. Para o HECOPS isto significa que a tabela de referência pode ter um limite de 10 linhas, ou seja, 10 referências e manter o mesmo nível de erro que uma tabela com mais de 10 linhas.



**Figura 29: Desempenho do HECOPS - média de erros de posição restringindo o número de referências na tabela.**

Uma comparação entre confiança na posição e erro desta é demonstrada na Figura 30. De acordo com Savarese, o Hop-Terrain apresenta um alto desvio padrão e portanto não possibilita outras análises [Savarese 2002]. Em contraste, o algoritmo HECOPS apresenta uma relação significativa entre confiança e erro como pode ser visto na Figura 30. Sensores com confiança maior que 0,65 têm um nível de erro médio abaixo de 10% e também apresentam baixo desvio padrão. Confianças menores que 0,65 equivalem a erros de aproximadamente 50% e alto desvio padrão. Isto comprova o conceito do algoritmo de que somente sensores com alta confiança nas suas próprias posições podem servir como referências para outros sensores calcularem suas próprias posições.

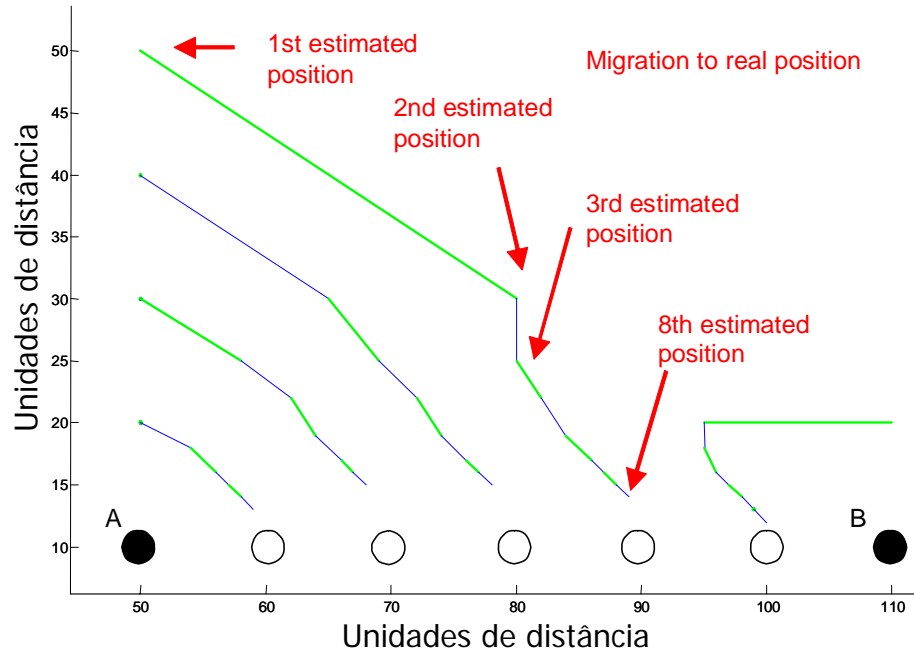


**Figura 30: Relação entre confiança e erro de posição (média e desvio padrão).**

Um último teste foi realizado para verificar o desempenho do algoritmo frente a um problema específico. Através de uma disposição particular dos sensores objetivou-se testar o comportamento multi-salto. Em outras palavras, quando as informações não são recebidas diretamente de um âncora, mas indiretamente de outros sensores. Estes outros sensores também recebem informação de forma indireta passada de sensor para sensor. Na

Figura 31 pode ser visto dois sensores âncoras (A e B) separados por uma fila de cinco sensores em linha reta. O alcance de cada sensor é limitado a 10. Portanto cada sensor pode se comunicar com no máximo dois vizinhos. A primeira estimativa de posição é grosseira como visto na

Figura 31. A cada iteração os sensores conseguem uma estimativa melhor. A conclusão é de que o HECOPS funciona em uma configuração multi-salto e com um mínimo de referências.



**Figura 31: Desempenho do HECOPS em um problema específico.**



## 5 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho de pesquisa tratou-se do problema de localização em RSSFs. Sua importância é fundamental para o desenvolvimento de aplicações que exploram as vantagens de uma RSSF. Destas vantagens destacam-se: coleta de dados e mapeamento de eventos em locais de difícil acesso, tratamento de um grande volume de medições e possibilidade de uso de algoritmos de roteamento geográficos.

Inicialmente foram apresentados alguns conceitos específicos do problema e logo após algumas soluções encontradas com uso de algoritmos. Em seguida foi apresentada e avaliada uma nova solução para o problema. Um algoritmo foi concebido através da observação do comportamento da propagação de ondas de radio frequência em círculos concêntricos em torno de um sensor que usa RSSI.

### 5.1 CONCLUSÕES

Nesta dissertação foi apresentado em detalhe um algoritmo distribuído para resolver o problema de localização de sensores dentro de uma rede *ad hoc* de sensores sem fio. O algoritmo apresentou duas principais inovações se comparado com outros existentes:

- 1) Usa menos referências para estimar sua posição;
- 2) Usa heurística para calibrar as medições de distância entre pares de sensores.

Conforme demonstrado pelas simulações descritas no capítulo 4, o uso de um menor número de referências diminuiu a propagação de erros comuns em outros algoritmos, resultado em melhor precisão. Além disto, diminuiu o esforço computacional tão importante quando referido a dispositivos mais simples de uma RSSF.

Dentre as heurísticas sugeridas, a principal e que foi testada no estudo, é comparar as medições de alcance de dois sensores e corrigir eventuais desvios pela informação

mais confiável. Esta heurística conseguiu ajudar a compensar os erros de propagação irregular de radio frequência causados pelo ambiente anisotrópico.

O desempenho do novo algoritmo foi testado através de simulações em vários cenários representando diferentes topologias de redes *ad hoc*. As simulações foram aplicadas nas mesmas bases também a um algoritmo existente (Hop-Terrain), e os resultados confrontados. É possível notar a partir de vários gráficos apresentados, que o HECOPS atinge menos de 10% de erros em cenários de 15% de medida de medição, população de 10% de âncoras e conectividade de 30 sensores.

Também foi possível fazer uma correlação entre o nível de confiança na posição e o erro efetivo da posição. Quando o sensor calcula para si uma confiança maior que 0,65, pode-se esperar erros em média menores que 10%. Este fato possibilita o estudo futuro de mais algoritmos, entre outros, o de roteamento baseados na posição geográfica.

## 5.2 TRABALHOS FUTUROS

Uma possível direção no sentido de estender o estudo é a implementação do algoritmo com experiências no mundo real. Os resultados práticos podem comprovar definitivamente se a heurística usada ajudaria na calibragem do sensor. Isto já está sendo feito no LISHA. Em uma primeira experiência utilizando-se de quatro sensores Tmote [Tmote 2006], dispostos na forma de quadrado tomou-se as leituras reais. Os primeiros resultados devem ser publicados logo. Notou-se até agora uma grande instabilidade na recepção utilizando-se RSSI.

Outro estudo importante seria o desempenho do algoritmo quanto ao tráfego de dados e energia na rede. Também seria muito interessante explorar a utilização em rastreamento de objetos móveis. A tecnologia RSSI tem um custo reduzido comparada com a ToA porém muito irregular para a utilização em rastreamento. A partir da calibragem baseado no HECOPS poder-se-ia tentar viabilizar esta utilização

## REFERÊNCIAS BIBLIOGRÁFICAS

- [Akyildiz 2002] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks”, *IEEE Communications Magazine*, 40(8):102-114, Agosto 2002.
- [Albowicz 2001] Albowicz, A. Chen, and L. Zhang, “Recursive position estimation in sensor networks,” in *Proc. IEEE Int. Conf. Network Protocols (ICNP’01)*, 2001, pp. 35–41.
- [Bahl 2000] P. Bahl and V. N. Padmanabhan. “Enhancements to the RADAR user location and tracking system”, Technical Report MSR-TR-2000-12, Microsoft Research, Fevereiro 2000.
- [Bahl2 2000] P. Bahl and V. N. Padmanabhan, “RADAR: An In-Building RF based User Location and Tracking System”, *Proceedings of IEEE INFOCOM 2000*, Tel-Aviv, Israel, Março 2000.
- [Boettcher 2002] P. Boettcher, J. A. Sherman, G. A. Shaw, “Target localization using acoustic time-difference of arrival in distributed sensor networks”, In *proceedings of SPIE 47th Annual Meeting*, 2002.
- [Biagioni 2002] E. Biagioni, and K. Bridges, “The application of remote sensor technology to assist the recovery of rare and endangered species”, In *Special issue on Distributed Sensor Networks for the International Journal of High Performance Computing Applications*, Vol. 16, N. 3, Agosto 2002.
- [Bulusu 2000] N. Bulusu, J. Heidemann, and D. Estrin, “GPS-less low cost outdoor localization for very small devices”, *IEEE Personal Communications*, 7(5):28 -34, Outubro 2000.
- [Caffery 1998] J.J. Caffery, G.L. Stber, "Overview of Radiolocation in CDMA Cellular Systems", *IEEE Communications Mag.*, Vol. 36, No. 4, Abril 1998, pp. 38-45.
- [Doherty 2001] Doherty, L., Pister, K., and Ghaoui, L. “Convex position estimation in wireless sensor networks” In *Proc. IEEE INFOCOM (Abril 2001)*.
- [Estrin 2001] Estrin, D; Culler, D.; Pister, K. e Sukhatme, G.,”Connecting the Physical World with Pervasive Networks”, *IEEE Pervasive Computing*, 1(1):59-69, 2002.

- [Ganesan 2002] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin and S. Wicker, "Complex Behavior at Scale: An Experimental Study of Low-Power Wireless Sensor Networks", Technical Report UCLA/CSD-TR 02-0013, 2002.
- [Girod 2001] L. Girod, D. Estrin, "Robust range estimation using acoustic and multimodal sensing", in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Maui, Hawaii, 2001.
- [Niculescu 2001] D. Niculescu, D., AND Nath, B., "Ad-hoc positioning system", In Proceeding of IEEE Global Communications Conference (GLOBECOM), pp.2926-2931, Novembro 2001.
- [Hightower 2001] J. Hightower and G. Borriello, "Location systems for ubiquitous computing", Computer, 34(8), 2001. IEEE Computer Society Press.
- [Howden 2002] E. Howden, "Networked sensors for the objective force", In proceedings of SPIE 47th Annual Meeting, 2002.
- [Niculescu 2003] D. Niculescu, D., AND Nath, B. "Ad Hoc Positioning System (APS) Using AOA". In Proc. of IEEE INFOCOM (Salt Lake City, UT, Abril 2003), pp. 2037–2040.
- [Ji 2004] X. Ji and H. Zha. "Sensor positioning in wireless ad-hoc sensor networks with multidimensional scaling". In Proceedings of the 23rd Conference of the IEEE Communications Society (IEEE INFOCOM), 2004.
- [Whitehouse 2005] Kamin Whitehouse, Alec Woo, Chris Karlof, Xiaofan Jiang, David Culler, "The Effects of Ranging Noise on Multi-hop Localization: An Empirical Study", In IPSN '05, Los Angeles, California, USA, Abril 25-27 2005. UCLA.
- [Karalamar 2006] Tufan Coskun Karalar , Tese: Implementation of a Localization System for Sensor Networks, University Of California, Berkeley, 2006
- [Kahn 1999] J.M. Kahn, R.H. Katz, K.S.J. Pister, "Next century challenges: Mobile networking for 'smart dust'", Proc. MOBICOM, 1999, Seattle, 271-278.
- [Langendoen 2003] K. Langendoen and N. Reijers., "Distributed localization in wireless sensor networks: a quantitative comparison", Computer Networks: The International Journal of Computer and Telecommunications Networking, Special issue: Wireless sensor networks, pages 499--518, Novembro 2003
- [Lorincz 2004] K. Lorincz and M. Welsh, "A Robust, Decentralized Approach to RF-Based Location Tracking", tech. report TR-19-04, Division of Eng. and Applied Sciences, Harvard Univ., 2004.
- [Matlab 2006] MATLAB® 7.0 - The Language of Technical Computing. Disponível em: <<http://www.mathworks.com/products/matlab/>>. Acesso em: janeiro 2006.

- [Nasipuri 2002] Nasipuri, A and K. Li, “A directionality based location discovery scheme for wireless sensor networks,” in Proc. ACM Int. Workshop on Wireless Sensor Networks and Applications (WSNA’02), 2002, pp. 105–111.
- [Oliveira 2005] H.A.B.F. Oliveira. Um algoritmo Recursivo de Localização para Redes de Sensores Sem Fio. Dissertação (Mestrado em Ciência da Computação) – Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de Minas Gerais, Belo Horizonte, Setembro, 2005.
- [Savarese 2002] C. Savarese, C., Rabaey, J., And Langendoen, K.,”Robust Positioning Algorithms for Distributed Ad-Hoc Wireless Sensor Networks”, In USENIX Annual Technical Conference, 2002.
- [Savarese 2001] C. Savarese, J. Rabaey, and J. Beutel, “Locationing in distributed ad-hoc wireless sensor networks”, In IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), pages 2037-2040, Salt Lake City, UT, Maio 2001.
- [Savvides 2001] A. Savvides, A., Han, C., and Srivastava, M.,”Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors”, In Proc. 7th ACM MOBICOM Conf. (Julho 2001), pp. 166–179.
- [Savvides 2002] A. Savvides, H. Park and M. Srivastava, “The Bits and Flops of the N-Hop Multilateration Primitive for Node Localization Problems”, In First ACM International Workshop on Wireless Sensor Networks and Application, Atlanta, GA, Setembro 2002.
- [Schwiebert 2001] L. Schwiebert, S. K. S. Gupta, and J. Weinmann, “Research challenges in wireless networks of biomedical sensors”, In Mobile Computing and Networking, páginas 151-165, 2001.
- [Slijepcevic 2002] S. Slijepcevic, S. Megerian, and M. Potkonjak,”Location errors in wireless embedded sensor networks: Sources, models, and effects on applications”, ACM Sigmobile Mobile Comp. & Commun. Review, 6(3):67--78, Julho 2002
- [Tmote 2006] Tmote sky. Disponível em:  
<<http://www.moteiv.com/products/tmotisky.php>>. Acesso em: janeiro 2006.
- [Vandenbussche 2005] Katelijne Vandenbussche , Tese: Fine-grained Indoor Localisation using Wireless Sensor Networks, Delft University of Technology, Delft, Agosto 2005.
- [Wang 2003] H. Wang, J. Elson, L. Girod, D. Estrin, and K. Yao, “Target classification and localization in habitat monitoring”, In Proceedings of the IEEE ICASSP 2003,Hong Kong , Abril 2003.
- [Wu 2005] Hongyi Wu, Chong Wang, Nian-Feng Tzeng, “Novel Self-Configurable Positioning Technique for Multihop Wireless Networks”, IEEE/ACM Transactions On Networking, Vol. 13, No. 3, Junho 2005, pp. 609-621.

# Livros Grátis

( <http://www.livrosgratis.com.br> )

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)  
[Baixar livros de Literatura de Cordel](#)  
[Baixar livros de Literatura Infantil](#)  
[Baixar livros de Matemática](#)  
[Baixar livros de Medicina](#)  
[Baixar livros de Medicina Veterinária](#)  
[Baixar livros de Meio Ambiente](#)  
[Baixar livros de Meteorologia](#)  
[Baixar Monografias e TCC](#)  
[Baixar livros Multidisciplinar](#)  
[Baixar livros de Música](#)  
[Baixar livros de Psicologia](#)  
[Baixar livros de Química](#)  
[Baixar livros de Saúde Coletiva](#)  
[Baixar livros de Serviço Social](#)  
[Baixar livros de Sociologia](#)  
[Baixar livros de Teologia](#)  
[Baixar livros de Trabalho](#)  
[Baixar livros de Turismo](#)