

# **Vírus: Localização Colaborativa para Sistemas de Posicionamento Interiores**

**Carla Margarida da Silva Marreiros**

Dissertação para obtenção do Grau de Mestre em

**Engenharia Eletrotécnica e de Computadores**

Orientadores: Prof. João Pedro Castilho Pereira Santos Gomes  
Prof. João Paulo Salgado Arriscado Costeira

**Júri**

Presidente: Prof. João Fernando Cardoso Silva Sequeira  
Orientador: Prof. João Pedro Castilho Pereira Santos Gomes  
Vogal: Marko Beko

**Novembro 2020**



# Declaração

Declaro que o presente documento é um trabalho original da minha autoria e que cumpre todos os requisitos do Código de Conduta e Boas Práticas da Universidade de Lisboa.

# Agradecimentos

Quero agradecer à minha família e aos meus amigos "de todas as ocasiões", como lhes costumo chamar, na esperança que esqueçam o leitão que prometi a todos :)

Um especial obrigado também ao Professor João Pedro Gomes pela sua paciência e disponibilidade.

Os falares emotivos não são o meu forte mas deixo-vos aqui o meu sincero Obrigado!

# Resumo

O trabalho desenvolvido nesta dissertação tem como ponto de partida um sistema de localização *in-door* desenvolvido pela Thales [1]. Este utiliza sensores baseados no *Open Beacon* [2] e apoia-se em medições de Received Signal Strength (RSS). Este sistema já foi alvo de estudo numa dissertação precedente a esta [3]. Naquela, foram descritas as metodologias utilizadas originalmente pela Thales e propostas outras que provaram ter melhores resultados, tendo por base um algoritmo de otimização convexa.

No sistema original (proposto pela Thales) aplicou-se às medições de RSS um Moving Average Filter (MAF) seguido da Fórmula de Friis. Tal mostrou-se insuficiente para determinar a localização correta de um alvo (antena móvel). Para isso, nesta dissertação é estudado um método diferente dos já utilizados, que também tem como ponto de partida as medições de RSS entre todos os pares de sensores. Dado que o processamento destas é feito *offline* e não em tempo real, o pretendido não foi a localização exata do alvo mas sim o traçado da trajetória que este descreveu ao longo do tempo.

Recorreu-se a metodologias como filtros inspirados nos de Kalman, e outros, ao algoritmo de Multidimensional Scaling (MDS) e análises *Procrustes*. Outros processos foram descartados no decorrer do trabalho desenvolvido, uma vez que provaram ser desnecessários e/ou inadequados.

Para a obtenção dos dados, medições de RSS, foram realizadas diferentes montagens, *setups*, das âncoras em vários cenários físicos diferentes. A montagem escolhida para dar a conhecer as metodologias escolhidas foi uma das que proporcionou melhores resultados, mostrando assim as melhores características para este tipo de operação.

## Palavras Chave

RSS; Medições de Potência; Redes de Sensores; Localização Colaborativa; Algoritmo; MDS

# Abstract

The work done during this dissertation has as starting point an indoor localization system developed by Thales [1]. It uses Open Beacon [2] based sensors and relies on RSS measurements. This system was already the subject of study of a previous dissertation [3]. In that one, the methodologies originally used were described and others were proposed that proved to achieve better results, based on a convex optimization algorithm.

In the original system (proposed by Thales) a Moving Average Filter(MAF) was applied to the measurements of RSS followed by the Friis Formula. This proved to be insufficient to determine the correct location of a target (mobile antenna). For this, in this dissertation a different method from the ones already used is studied, which also has as starting point the RSS measurements between all pairs of sensors. Given that the processing of these is done offline and not in real time, what was intended was not the exact location of the target, but the outline of the trajectory it described over time.

Among the used methodologies, there are filters resembling the Kalman ones and others, MDS algorithm and Procrustes Analysis. Other processes were discarded as they proved to be unnecessary and/or unsuitable.

To obtain the needed data (RSS measurements) many setups of the anchors in several different physical scenarios were examined. The selected setup to illustrate the chosen methodologies was one of the ones that produced the best results, showing the preferred features to this kind of operation.

## Keywords

RSS; Power Measurements; Wireless Sensor Networks; Cooperative Localization; Algorithm; Multidimensional Scaling

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>2</b>
1.1	Motivação . . . . .	3
1.2	Objetivos . . . . .	4
1.3	Redes de Sensores Wireless . . . . .	4
1.3.1	Experiência 'Vírus' . . . . .	5
1.4	Contribuições . . . . .	5
<b>2</b>	<b>Revisão da Literatura</b>	<b>7</b>
2.1	Tipos de Sinais . . . . .	8
2.1.1	Acústicos . . . . .	8
2.1.1.A	Ultrassom . . . . .	8
2.1.2	Infravermelhos . . . . .	8
2.1.3	Óticos . . . . .	9
2.1.4	Rádio-Frequência . . . . .	9
2.1.4.A	RFID . . . . .	9
2.1.4.B	Ultrawideband . . . . .	10
2.1.4.C	Wi-Fi . . . . .	10
2.1.4.D	Bluetooth . . . . .	10
2.2	Métricas . . . . .	11
2.2.1	Angle of Arrival . . . . .	11
2.2.2	Time of Flight . . . . .	11
2.2.3	Received Signal Strength . . . . .	11
2.3	Multipath e Fading . . . . .	12
2.3.1	Multipath . . . . .	13
2.3.2	Fading . . . . .	13
2.4	Algoritmos . . . . .	14
2.4.1	Range-based . . . . .	14
2.4.1.A	Geométricos . . . . .	14

2.4.1.B	Optimization-Based . . . . .	15
2.4.2	Algoritmos Range-Free . . . . .	16
2.4.2.A	Fingerprinting . . . . .	17
<b>3</b>	<b>Metodologias</b>	<b>18</b>
3.1	Multidimensional Scaling . . . . .	20
3.1.1	MDS Não-Métrico . . . . .	21
3.1.2	MDS Clássico vs MDS Iterativo . . . . .	22
3.1.3	SMACoF . . . . .	23
3.1.4	MDSCALE - MATLAB . . . . .	27
3.2	Análise Procrustes . . . . .	28
<b>4</b>	<b>Resultados</b>	<b>31</b>
4.1	Trabalho Experimental . . . . .	33
4.1.1	Sensores (Tags) . . . . .	33
4.1.2	Experiência Thales . . . . .	33
4.1.2.A	Comunicação entre tags . . . . .	34
4.1.3	Montagens . . . . .	35
4.1.3.A	Cenário Cooperativo . . . . .	35
4.1.3.B	Disposição espacial e ambientes . . . . .	35
4.1.3.C	Obtenção das medições de RSS . . . . .	36
4.2	Resultados Preliminares . . . . .	36
4.2.1	Orientação dos sensores . . . . .	38
4.2.2	Potência vs Distância . . . . .	38
4.2.3	Preparação das potências . . . . .	39
4.2.3.A	Regressão Linear . . . . .	39
4.2.3.B	Regressão Linear (Pares) . . . . .	41
4.3	Resultados . . . . .	42
4.3.1	Limpeza, Completamento e Balizagem . . . . .	42
4.3.1.A	Limpeza e Completamento . . . . .	42
4.3.1.B	Balizagem . . . . .	44
4.3.2	Filtro de Kalman . . . . .	46
4.3.3	Entrada do MDS . . . . .	48
4.3.4	MDSCALE VS SMACoF . . . . .	48
4.3.4.A	Efeito da Análise Procrustes . . . . .	49
4.3.4.B	Impacto da Adição do Target . . . . .	50
4.3.4.C	Análise Procrustes - Target . . . . .	51



4.3.4.D	Filtragem das distâncias . . . . .	52
4.3.5	MDSCALE - Efeitos do ambiente . . . . .	53
4.3.5.A	Átrio do Campus TagusPark . . . . .	54
4.3.5.B	Garagem do Campus TagusPark . . . . .	56
<b>5</b>	<b>Conclusão</b>	<b>59</b>
5.1	Apreciação Global . . . . .	61
5.1.1	Conclusão . . . . .	62
5.1.2	Trabalho futuro . . . . .	62

# Lista de Figuras

2.1 Reflexão ( <i>Reflection</i> ), difração ( <i>Diffraction</i> ) e <i>scattering</i> [4] . . . . .	13
2.2 Triangulação com três âncoras, posicionadas em $p_1$ , $p_2$ e $p_3$ , e alvo localizado em $p_0$ [5] .	15
3.1 Distância versus Potência. . . . .	19
3.2 Tabela de correlações e gráfico correspondente [6] . . . . .	20
3.3 Princípio de um algoritmo iterativo que utilize MDS. . . . .	23
4.1 Sensores utilizados. [3] . . . . .	33
4.2 OpenBeacon estação-base (EasyReader). . . . .	34
4.3 Esquema de comunicação entre tags [3]. . . . .	34
4.4 Troca de pacotes entre tags e Base Station (BS) [3]. . . . .	35
4.5 Disposição das âncoras - DSOR. . . . .	37
4.6 Coordenadas das âncoras, com origem na tag 4. . . . .	37
4.7 Trajetória descrita pelo <i>target</i> , entre as âncoras (de 1 a 5). . . . .	38
4.8 Diagrama de propagação da antena de um sensor (retirada de [3]). . . . .	38
4.9 Em cima: reta de regressão linear. Em baixo: representação dos resíduos da reta de regressão. . . . .	40
4.10 "Reta de regressão linear". . . . .	42
4.11 Gráficos $P(d)$ antes (a) e depois (b) da aplicação do algoritmo Limpeza, Completamento e Balizagem (LCB) (Algoritmo 4.1) - eixo horizontal: tempo(s); eixo vertical: potência(dBm)	45
4.12 Medições entre as âncoras 1 e 3 antes e depois da aplicação do filtro de Kalman Aproximado. . . . .	48
4.13 Representação das âncoras - algoritmo MDSCALE. . . . .	49
4.14 Representação das âncoras - algoritmo SMACoF. . . . .	50
4.15 Âncoras - algoritmo MDSCALE. . . . .	51
4.16 Âncoras - algoritmo SMACoF. . . . .	51
4.17 Posições do <i>target</i> obtidas pelo algoritmo MDSCALE. . . . .	52

4.18	Posições do target obtidas pelo algoritmo SMACoF. . . . .	52
4.19	Trajetórias do Target. . . . .	54
4.20	Montagem no átrio do TagusPark. . . . .	55
4.21	Distância <i>versus</i> Potência - montagem h04. . . . .	56
4.22	Coordenadas das âncoras - montagem no átrio do TagusPark. . . . .	56
4.23	Trajetória descrita pelo <i>target</i> (MDSCALE) - átrio do TagusPark. . . . .	57
4.24	Montagem na garagem do TagusPark. . . . .	57
4.25	Coordenadas das âncoras - montagem na garagem do TagusPark. . . . .	58
4.26	Trajetória descrita pelo <i>target</i> (MDSCALE) - garagem do TagusPark. . . . .	58

# Lista de Tabelas

4.1	Métricas/indicadores e respetivos valores. . . . .	41
4.2	Métricas/indicadores e respetivos valores. . . . .	42
5.1	Ambientes onde foram realizadas as montagens descritas nesta dissertação. . . . .	61

# Acrónimos

<b>RANSAC</b>	RANdom SAmples Consensus
<b>MM</b>	Majorization-Minimization
<b>NMDS</b>	Non Metric Multidimensional Scaling
<b>MAF</b>	Moving Average Filter
<b>TOA</b>	Time of Arrival
<b>LCB</b>	Limpeza, Completamento e Balizagem
<b>RMSE</b>	Root Mean Square Error
<b>BS</b>	Base Station
<b>LED</b>	Light Emitting Diode
<b>SoC</b>	System On a Chip
<b>SVD</b>	Singular Value Decomposition
<b>SSE</b>	Sum Squared Error
<b>CG</b>	Gradiente Conjugado
<b>SMACoF</b>	Scaling by MAjorizing a COMplicated Function
<b>PAVA</b>	Pool-Adjacent-Violators Algorithm
<b>MDS</b>	Multidimensional Scaling
<b>LLSQ</b>	Linear Least Squares
<b>ML</b>	Maximum Likelihood
<b>SVM</b>	Support Vector Machines
<b>GPS</b>	Global Positioning System
<b>WSN</b>	Wireless Sensor Network
<b>AoA</b>	Angle of Arrival
<b>AP</b>	Access Point

<b>BLE</b>	Bluetooth Low Energy
<b>CL</b>	Computador Local
<b>ISM</b>	Industrial, Scientific and Medical Band
<b>IV</b>	Luz Infravermelha
<b>LoS</b>	Linha de Vista
<b>NLoS</b>	Sem Linha de Vista
<b>PLE</b>	Path Loss Exponent
<b>PL</b>	Path Loss
<b>RF</b>	Rádio-Frequência
<b>RFID</b>	<i>Radio Frequency Identification</i>
<b>RSS</b>	Received Signal Strength
<b>TOF</b>	Time of Flight
<b>UWB</b>	Ultrawideband
<b>WLAN</b>	Wireless Local Area Network

# 1

## Introdução

### Conteúdo

---

1.1	Motivação . . . . .	3
1.2	Objetivos . . . . .	4
1.3	Redes de Sensores Wireless . . . . .	4
1.4	Contribuições . . . . .	5

---

Um sistema de localização/posicionamento *indoor* é um sistema que consegue determinar a posição de um objeto ou pessoa, num determinado sítio (limitado a um espaço). Estes sistemas são requisitados por redes de sensores *wireless*, *Wireless Sensor Networks (WSNs)*.

Ao contrário do que acontecia até pouco mais do que uma década atrás, quando um utilizador pretende conhecer a sua própria localização, a de uma rua ou de um edifício, basta recorrer a um dispositivo móvel, ou não, como um *laptop* ou *smartphone*. Já no dispositivo, é possível aceder a aplicações, como a tão conhecida *Google Maps*, que fornecem a informação pretendida. Estas aplicações, por norma, recorrem ao mais popular sistema de localização, o *Global Positioning System (GPS)*, que foi projetado para cenários *outdoor*. Assim, se alguém se encontrar perdido dentro de um grande museu e quiser localizar-se, é inútil recorrer às referidas aplicações. Para evitar estas situações, algumas instalações têm ao dispor dos visitantes sistemas de posicionamento, como o *Wifarer* [7].

Desde que foi generalizado para uso civil, com erros de localização menores que 10 metros - condição inaceitável indoor - o GPS tornou-se num sistema bastante valioso, tanto para efeitos de localização como para navegação, veicular e pedestre. O facto de ter uma precisão satisfatória em *outdoor* [8] e se encontrar em tecnologias orientadas ao consumidor (como o dispositivo móvel que possui o mesmo nome) contribuíram para a sua utilização em larga escala. Porém, apesar do seu inegável valor, este sistema não se encontra adaptado para certos ambientes, como o *indoor* e o aquático, produzindo dados errados ou chegando mesmo a não funcionar de todo. Como efetua a localização de dispositivos através do tempo de voo, Time of Flight (TOF), de sinais rádio transmitidos por uma constelação de satélites, em ambientes onde a vista do céu se encontra obstruída, é impossível obter a localização desta forma.

Outros sistemas que surgiram como alternativa e/ou complemento do GPS também apresentam as suas desvantagens, como é o caso do *Skyhook Wireless* [9]. Este não recorre a sinais de Rádio-Frequência (RF) enviados por satélites, como o GPS, mas sim a sinais Wi-Fi fornecidos por vários Access Points (APs) Wi-Fi (que podem pertencer tanto a um estabelecimento comercial como a uma habitação particular), a partir dos quais é construída uma enorme base de dados e feito o mapeamento (*fingerprinting*) de uma cidade, por exemplo. Este processo é complexo e demorado, uma vez que é necessária a construção de base de dados muito grande, que está incessantemente em renovação, pois os APs estão sempre sujeitos a mudança (de sítio e/ou substituição dos equipamentos).

## 1.1 Motivação

Atualmente, quer no campo da tecnologia quer na sociedade em geral, ser possível localizar, em tempo real, uma pessoa ou um objeto (ou várias/os) não só é algo desejável mas também absolutamente necessário, em casos relacionados com aplicações médicas e segurança. Seria fantástico uma



enfermeira olhar para o seu smartphone e ver em que sala se encontra certo equipamento, ou a central de segurança de uma exposição valiosa poder monitorizar cada um dos seus visitantes, de forma eficiente.

Com o aumento das dimensões e da complexidade das infraestruturas, nasceu no meio tecnológico a necessidade de produzir um sistema capaz de efetuar localização indoor e com uma precisão adequada: o erro de estimação deve ser tão pequeno quanto possível (idealmente, menor que 1 metro). Este sistema deve também utilizar tecnologias já existentes e/ou que permitam interoperabilidade entre os vários dispositivos, bem como ser orientado ao consumidor, isto é, de fácil uso. Outro aspeto a ter em conta é o custo da implementação; com esta preocupação em vista é importante referir que o sistema a ser estudado é de baixo custo.

## 1.2 Objetivos

O principal objetivo desta dissertação é o desenvolvimento de um método para a obtenção da localização de um sensor móvel - *target* - a partir das medições de RSS entre este e as âncoras - sensores colocados de forma estratégica no espaço em estudo e que se encontram imóveis. Pretende-se tornar o sistema de localização o mais eficiente possível, mantendo os mesmos sensores, disponibilizados pela Thales. Dada a fraca qualidade das medições, o pretendido não foi a localização exata do target em cada instante, mas sim o traçado da trajetória descrita por si ao longo do tempo.

Após uma extensa pesquisa sobre o tema das redes de sensores, foi feito um "estado da arte" sobre as várias métricas recolhidas/analizadas nestas redes e os algoritmos mais utilizados em sistemas de localização. Foram averiguadas as técnicas matemáticas e linhas de pensamento disponíveis e passíveis de serem utilizadas no desenvolvimento do trabalho. Simultaneamente, realizou-se um certo número de montagens, com os sensores dispostos de forma estratégica, o que se resumiu à componente prática neste estudo. Por fim, foi idealizada e concretizada a versão final do método referido acima, apresentado nesta dissertação.

## 1.3 Redes de Sensores Wireless

Uma WSN é uma rede na qual os sensores (que também podem ser designados por *tags*) se encontram espacialmente distribuídos e comunicam entre si, se a rede for cooperativa, ou apenas com o Computador Local (CL) [10]. Este último, o CL, é necessário, pelo menos, para fazer a ponte entre os sensores e o utilizador, mas também pode ser utilizado para fazer os cálculos necessários (que dependem dos algoritmos escolhidos). O espaço onde a rede se encontra implementada pode ter quaisquer dimensões: desde uma única sala a um edifício inteiro.

O objeto a ser localizado/posicionado é chamado de alvo (*target*), enquanto que os sensores/*tags* que constituem a rede são denominados nós ou âncoras (*anchors*), se a sua localização for conhecida *a priori*. Todos os sensores efetuam algum ou alguns tipos de medições.

Ao contrário do que acontece com redes constituídas com dispositivos que suportem GPS, os sensores utilizados nas WSNs são de baixo custo e de montagem fácil, embora o tempo de vida das suas baterias possa ser reduzido, o que é indesejável. Para além das anteriormente referidas, outra vantagem das WSNs é a mobilidade, que para sistemas *indoor* tem um papel chave. Esta permite aos sensores detetarem fenómenos de mudança, como mudanças de mobília; permite também a existência de alvos dinâmicos, que se encontram em movimento (*targets*) - localização dinâmica.

### 1.3.1 Experiência 'Vírus'

A base do trabalho desenvolvido nesta dissertação é uma atividade experimental realizada pela Thales [1] para a exposição "Viral", que teve lugar no Pavilhão do Conhecimento em 2015/16. A sala de exposições estava dividida por cinco áreas, nas quais se encontravam várias âncoras espalhadas estrategicamente e vários *targets*, carregados por pessoas que circulavam livremente no espaço, que efetuavam medições de RSS (entre eles e os outros sensores) e reportavam a um CL. O objetivo da atividade era simular ao vivo o fenómeno de propagação dum vírus entre os visitantes, mas veio a verificar-se que as deficiências na localização dos dispositivos transportados pelos utilizadores dificultavam o acompanhamento da actividade na consola de visualização instalada na sala.

No presente trabalho, cada um dos sensores mede o valor de RSS, que é o valor da potência com que o sinal transmitido chega ao recetor, isto é, cada *tag* reporta ao CL o valor de potência que recebeu de cada uma das outras *tags*. Este tipo de montagem é de grande interesse pois estes *beacons* permitem uma maior cobertura do que outras soluções, uma vez que são mais baratos e, com o mesmo orçamento de outra solução, poderão ser implantados mais sensores no mesmo espaço.

## 1.4 Contribuições

Primeiramente, é apresentado o estado-da-arte sobre sistemas de posicionamento. São enumerados os tipos de sinais utilizados e as principais métricas que se obtêm a partir destes, bem como os algoritmos mais conhecidos neste campo nos quais os sistemas de localização se costumam apoiar.

A principal contribuição desta dissertação prende-se com a criação de um método que permite determinar o posicionamento de um *target*, ao longo do tempo. Este método utiliza um algoritmo MDS, criado no seio da comunidade da Psicologia. Após pesquisa na literatura disponível, verificou-se que o MDS ainda é pouco utilizado noutros campos mas existem já vários trabalhos relacionados com localização que recorrem a este algoritmo como estes em [11] [12] [13]. Tal não é de estranhar pois, o

MDS é um algoritmo acessível com algumas variações, tendo-se utilizado uma das suas mais versáteis: Non Metric Multidimensional Scaling (NMDS). Para além deste, no método proposto figuram mais metodologias.

Durante a fase experimental deste estudo foram realizados vários *setups* em vários ambientes (vários espaços físicos). Tal estratégia permitiu averiguar o efeito que diferentes tipos de estruturas e disposições das âncoras e demografia do espaço têm na qualidade dos resultados.

Os resultados obtidos foram satisfatórios e, fazendo o balanço de todo o trabalho desenvolvido, a autora desta dissertação considerou-a uma adição positiva ao tema da localização *indoor*.

# 2

## Revisão da Literatura

### Conteúdo

---

2.1	Tipos de Sinais . . . . .	8
2.2	Métricas . . . . .	11
2.3	Multipath e Fading . . . . .	12
2.4	Algoritmos . . . . .	14

---

Neste capítulo será feita a revisão da literatura, ou estado da arte, acerca das WSNs. Serão brevemente analisados os tipos de sinais que podem ser utilizados neste tipo de redes, as medições que costumam ser efetuadas e os algoritmos de localização mais populares.

## 2.1 Tipos de Sinais

Nos últimos anos, o aumento do interesse em tecnologias que permitam mobilidade aos utilizadores tem potenciado a criação de dispositivos equipados com tecnologias *wireless*. Este fenómeno levou a que várias empresas, assim como instituições académicas, investissem no desenvolvimento de sistemas de localização que se baseassem em tecnologias *wireless*.

Nesta secção, o principal foco incidirá sobre os sinais RF, mais concretamente na medição da sua RSS. Contudo, serão enumerados e brevemente descritos outros tipos de sinais e medições.

### 2.1.1 Acústicos

#### 2.1.1.A Ultrassom

O sinal de ultrassom é uma onda de pressão sonora oscilante, com uma frequência maior do que o máximo audível pelo ser humano. Os dispositivos que utilizam estas ondas podem operar em frequências desde dezenas de kHz até cerca de 1 MHz [14]. Estes sinais têm uma baixa velocidade de propagação, quase não penetram em obstáculos (paredes, etc.) e os seus transdutores são de baixo custo, geralmente.

A grande vantagem dos sinais acústicos é a medição do TOF ser fácil e barata. No caso de ser conseguida a sincronização temporal de todos os nós constituintes da rede em questão, é possível alcançar-se uma precisão de poucos centímetros, com recurso a medições de TOF. A aplicação dos sinais de ultrassom em sistemas de localização *indoor* pode ser encontrada em sistemas como o *Active Bat* [15] e *Cricket* [16].

### 2.1.2 Infravermelhos

A Luz Infravermelha (IV) é radiação eletromagnética com comprimentos de onda maiores do que os da luz visível (dos 700 nm a 1 mm), ou seja, com frequências menores. Sistemas de posicionamento baseados nestas ondas apresentam uma boa precisão, porém é necessário que o emissor e o recetor estejam sempre em Linha de Vista (LoS) e sem interferências de fontes de luz.

Nestes sistemas, os recetores de IV são distribuídos estrategicamente num espaço (um edifício, por exemplo), com o objetivo de detetarem a presença <sup>1</sup> [17] destes beacons e, assim, posicionarem os

---

<sup>1</sup> Por detetarem a presença entenda-se a leitura das fases das ondas.

mesmos no espaço.

### 2.1.3 Óticos

Baseando-se em imagens, os sistemas óticos de localização (ou *vision-based*) conseguem, fazendo uso de técnicas de processamento de imagem, fornecer informação acerca da posição de um objeto/-pessoa [18]. Os sensores, que aqui são cameras, podem ser transportados pelo objeto a ser localizado ou por um outro elemento e, comparando a informação obtida com a fornecida pelas âncoras (previamente instaladas no espaço), conseguem detetar certos fenómenos, como por exemplo se alguém está sentado numa determinada cadeira. Como se pode facilmente deduzir, pelo facto de se depender de informação visual, a base de dados destes sistemas precisa de muitas atualizações, já que qualquer mudança no ambiente, como trocas de mobília, podem afetar gravemente os resultados. Para além de serem afetados por muitas interferências, como a luz, estes sistemas apresentam outra desvantagem que é a falta de segurança/privacidade (pois é necessária a captura de imagens).

Devido a todo o equipamento e processos necessários para o funcionamento de sistemas óticos, estes tornam-se muito caros.

### 2.1.4 Rádio-Frequência

Tal como em qualquer sistema que efetue comunicações, também os que utilizam sinais rádio têm o objetivo de trocar informação entre um emissor e um recetor (pelo menos). Em condições ideais, este último receberia o sinal original. Na realidade, o sinal sofre atenuação, que resulta na perda de potência do sinal, ao longo do tempo. Nas subsecções seguintes encontram-se as tecnologias RF mais utilizadas em localização.

#### 2.1.4.A RFID

A tecnologia *Radio Frequency Identification* (RFID) é bastante utilizada, não tanto para sistemas de localização, mas para sistemas de deteção de proximidade [19], como é o caso da tecnologia presente nas cancelas do metro lisboeta.

Um sistema RFID consiste numa *tag*, um leitor e uma antena. A *tag* encontra-se incorporada nalgum objeto, como um cartão, que pretenda ser detetado. O leitor, que não é mais do que um transdutor (que emite energia para alimentar a tag RFID), é capaz de ler a informação fornecida pela *tag* próxima, i.e., a sua ID, a qual contém uma antena que permite a comunicação.

#### 2.1.4.B Ultrawideband

É uma tecnologia rádio para a transmissão de informação numa banda larga (maior que 500 MHz), que utiliza uma grande porção do espectro disponível [20]. Os sinais de Ultrawideband (UWB) têm curta duração, em relação aos relativos a outras tecnologias, o que torna possível filtrar os sinais originais dos refletidos e detetar os instantes de receção dos pacotes e, por conseguinte, uma melhor fiabilidade na medição de TOF e/ou Time of Arrival (TOA), o que resulta numa melhor precisão nos resultados. Outra característica apelativa destes sinais é a de conseguirem penetrar vários obstáculos, como roupa e paredes, embora líquidos e metais constituem fontes de interferências (o que pode ser ultrapassado pela colocação de mais sensores em áreas críticas).

#### 2.1.4.C Wi-Fi

As Wireless Local Area Networks (WLANs) de alcance médio, que operam na banda Industrial, Scientific and Medical Band (ISM) (dos 2.4 aos 2.485 GHz), têm, nos últimos anos, servido de apoio a esquemas de localização, com a adição de um CL. Com um alcance de até 100 m, o IEEE 802.11 é o *standard* dominante nestas redes [21].

Muitos lugares têm Wi-Fi disponível e qualquer dispositivo com uma antena Wi-Fi (*smartphones, tablets, etc.*) pode ser um nó da rede. Este nó comunica com um ou mais APs, e vice-versa, e regista as medidas de RSS efetuadas, enviando-as para o CL, que determina a localização do nó.

#### 2.1.4.D Bluetooth

Nos dias que correm, a presença da tecnologia *Bluetooth* em vários aparelhos móveis é algo muito comum. Para a transmissão de informação, o seu alcance é de cerca de 15 m e o seu *bit rate* é de cerca de 1 Mb/s.

O *Bluetooth* é um *standard* leve e flexível que suporta outros serviços para além de sistemas de localização. Contudo, por si só não é suficiente: devido ao seu alcance, o sistema não é preciso, pelo que não é viável considerar apenas proximidades entre sensores. É necessário então efetuar outras medições, como RSS. As antenas e, por conseguinte, as *tags* são dispositivos pequenos, o que torna a tecnologia *low cost*. Com a criação de novas aplicações de *healthcare*, segurança, entre outras, surgiu o Bluetooth Low Energy (BLE) [22], também conhecido como *Bluetooth smart*, que apresenta uma considerável redução no consumo de energia, continuando a operar na banda ISM, não licenciada. Por todas estas características, esta é uma tecnologia promissora.

## 2.2 Métricas

Para determinar a localização de todos os sensores da rede, os sistemas de localização recorrem a medições que os últimos efetuam. Estas medições baseiam-se em vários tipos de métricas, as quais são descritas nesta secção.

### 2.2.1 Angle of Arrival

O Angle of Arrival (AoA), ângulo de chegada [23], é o ângulo do sinal intercetado pelo recetor. Teoricamente, é possível medir-se o AoA a partir de duas antenas. A direção de um sinal pode ser calculada explorando e detetando a diferença de fase entre as antenas do agregado de antenas no recetor. O ângulo é calculado comparando a direção do sinal que chega com uma orientação de referência, dado que o recetor conhece a sua própria orientação - é uma âncora ou possui uma bússola.

### 2.2.2 Time of Flight

O TOF é o tempo decorrido durante a propagação de um sinal, desde que foi emitido até ao recetor. Após receber o sinal, cada recetor envia o tempo de chegada do respetivo sinal para o computador central, no caso de se tratar de uma rede centralizada. A sua característica mais desejável é a de que, se fosse possível medir-se com precisão e sincronismo, o cálculo de distância seria muito fácil: calcular o produto entre o TOF medido e a velocidade de propagação. Contudo, nas tecnologias orientadas ao consumidor existem muitas limitações quanto à medição de tempos, o que faz com que não exista sincronismo entre os elementos da rede. Por este motivo, criou-se outra métrica, o *two-way* TOF [24], cujos requisitos de sincronização não são tão restritos.

### 2.2.3 Received Signal Strength

A RSS é a potência com que o sinal emitido por uma fonte chega ao recetor. Pode ser medida por qualquer dispositivo que possua uma antena própria, sem precisar de hardware adicional. As observações são utilizadas para determinar a distância entre o objeto a ser localizado e um *beacon*. Em condições ideais, a distância pode ser calculada com base na relação entre a potência emitida e a recebida (sendo a emitida constante). Contudo, a RSS varia com o tempo devido ao desvanecimento por *multipath* e outros fenómenos, sobretudo em cenários *indoor*, sendo que quanto maior a distância maior será a atenuação. Por esta razão, entre outras, é necessário filtrar os sinais recebidos, o que se prende com o objetivo do presente estudo.



## 2.3 Multipath e Fading

As ondas rádio, incluindo as de *Bluetooth*, não se mantêm constantes ao longo do tempo. Elas são influenciadas pelos objetos/pessoas que encontram no seu caminho, não seguindo, portanto, uma linha reta: quando uma onda colide com um objeto pode sofrer reflexão, o que leva à existência de vários caminhos de um emissor para um recetor – propagação *multipath* (multicaminho) [25]. Os valores de RSS medidos estão afetados por este fenómeno, o que pode resultar em atenuação ou amplificação do sinal original.

Para além de sofrerem *multipath*, as ondas, ao chocarem ou passarem através de objetos, incorrem em perda de energia e perda pontual de amplitude por interferência destrutiva entre reflexões, o que se chama de *fading*. Em adição aos problemas já referidos, as propriedades físicas do meio de propagação também influenciam os sinais. Por isto, é praticamente impossível desenhar um algoritmo capaz de prever todas as interferências e fenómenos sofridos pelo sinais durante a sua propagação, a começar pela grande complexidade das variáveis de entrada que seriam necessárias.

Quando o meio de propagação de uma onda é o ar e o emissor e recetor se encontram em LoS (propagação em *free space*), pode-se empregar a Fórmula de Friis para o cálculo da potência recebida (2.1), em dB

$$P_{R[dB]} = P_{E[dB]} + G_{E[dB]} + G_{R[dB]} - L_{[dB]} - \gamma 10 \log_{10} \left( \frac{\lambda}{4\pi d} \right) \quad (2.1)$$

na qual  $P_R$  e  $P_E$  correspondem à potência na entrada do recetor e à potência na saída do emissor, respetivamente,  $G_R$  e  $G_E$  aos ganhos das antenas,  $d$  à distância entre as mesmas,  $\lambda$  ao comprimento de onda do sinal,  $L$  a perdas alheias ao sinal e  $\gamma$  ao coeficiente de atenuação, Path Loss Exponent (PLE). Em cenários *outdoor* o PLE costuma ser igual a 2, sendo maior em cenários *indoor* (pois a atenuação é muito maior).

O sinal está a ser atenuado desde que saiu da fonte, o que resulta na perda de energia, e a potência perdida no caminho, Path Loss (PL), pode ser calculada a partir de (2.2), se o PLE for conhecido

$$PL_{[dB]} = PL_{0[dB]} + \gamma 10 \log_{10} \left( \frac{d}{d_0} \right) \quad (2.2)$$

onde  $d_0$  é a distância de referência para a qual PL é conhecida.

Contudo, como já referido, o sinal é fortemente afetado por *multipath* e *fading*, o que faz com que não se consiga saber o correto valor do PLE e não se possam utilizar as equações 2.1 e 2.2.

### 2.3.1 Multipath

A propagação *multipath* é causada por reflexão, difração e *scattering*, sobretudo em condições de Sem Linha de Vista (NLoS) entre o emissor e o recetor.

Nem sempre a reflexão é prejudicial para o sinal. Devido a esta é possível que um corredor se comporte como um guia de ondas e que a PL do sinal seja menor, em comparação com o valor obtido na propagação em espaço aberto. É evidente que, mesmo na condição do guia de ondas, o valor da PL nunca é nulo. Na verdade, apesar de a onda original e a refletida apresentarem os mesmos ângulos de chegada e partida, existe sempre perda de energia do sinal, e, por consequência disso, perda de potência. A energia da onda refletida depende não só do ângulo de partida mas também das propriedades do material refletor.

Quando uma onda rasa uma esquina, "curva-se" à sua volta e continua a trajetória. Dado que existe uma variedade tão extensa de objetos tão diferentes, é impossível modelar a difração com base num modelo teórico. Para este efeito, que é válido na maioria dos casos, adota-se o modelo de difração *Knife Edge* [26] que enuncia: uma obstrução a uma onda age como fonte secundária, fornecendo uma nova onda. Esta última propaga-se na área de sombra do obstáculo.

Outro fenómeno que leva à ocorrência de *multipath* é *scattering* [27]. Este acontece quando a onda choca com partículas que se encontravam no meio ou quando colide com uma superfície não lisa. As ondas resultantes de *scattering* apresentam muito menos energia que a onda original.

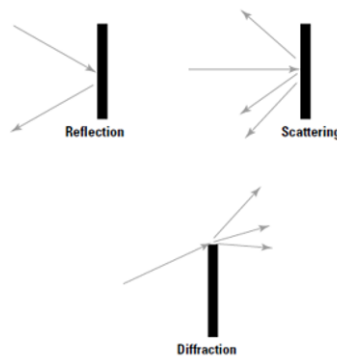


Figura 2.1: Reflexão (*Reflection*), difração (*Diffraction*) e *scattering* [4]

### 2.3.2 Fading

O tipo de *fading* experienciado por um sinal, que se propaga num canal, depende da natureza do sinal e das características do canal. Existem dois tipos de fading: *slow* e *fast*.

O *slow fading* prende-se com o facto de o emissor e o recetor se encontrarem em NLoS. Qualquer obstáculo que bloqueie o caminho direto, da onda original, adiciona atenuação, para além da atenuação

causada pelo meio e que é dependente das propriedades deste último. Designada por *slow fading*, esta PL adicional também é comumente denominada *shadowing*.

Já o *fast fading*, surge devido à interferência entre as várias versões da onda transmitida, criadas por fenómenos de *multipath*, que chegam ao recetor em instantes de tempo diferentes e com amplitudes e fases diferentes também. Tipicamente, o *fast fading* impõe-se sobre o *slow fading* e pode ser modelado utilizando uma distribuição de Rice ou de Rayleigh, dependendo se existe ou não um caminho direto entre o transmissor e o recetor, respetivamente.

## 2.4 Algoritmos

A deteção de sinais (Secção 2.1) ou mesmo a obtenção de certas métricas (Secção 2.2) a partir destes, não é suficiente para a estimação de distâncias entre nós e posterior localização. São necessários cálculos e, para tal, recorre-se a algoritmos, que podem ser *range-based* ou *range-free* e serão brevemente descritos nas secções seguintes.

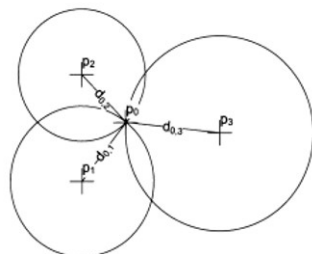
### 2.4.1 Range-based

Os algoritmos *range-based* fazem uso das medições adquiridas, através dos sensores, para estimar a distância entre os nós e posterior localização. Podem-se utilizar diferentes técnicas, como a trilateração (ou triangulação, quando se utilizam direções de chegada, AoA), acompanhadas de outras que melhorem a precisão da localização [28]. Ao contrário das soluções *range-free*, as *range-based* não necessitam de configurações *on-site* por parte do utilizador. Deste modo, o seu processo de instalação é fácil e de tempo reduzido, face às outras, pois consiste apenas na disposição espacial dos *beacons/tags/sensores*. Outra característica destas soluções é o facto de serem, à partida, auto-adaptativas: mesmo sem preverem alterações no ambiente conseguem fornecer a localização com a mesma (ou aproximadamente a mesma) precisão.

#### 2.4.1.A Geométricos

Um dos algoritmos *range-based* geométricos mais utilizados é a Trilateração [29]. As medições efetuadas, nomeadamente as de RSS, podem ser traduzidas para distâncias através de um modelo de propagação. Com a estimação da distância de um nó  $a$ , pelo menos, 3 outros nós que não sejam colineares e cuja posição seja conhecida, é possível descobrir, em  $\mathbb{R}^2$ , a localização do nó: a interseção entre as circunferências de centro em cada *beacon* e de raio igual à respetiva distância estimada (Figura 2.2). Devido à variação da RSS, impossível de controlar, as medições obtidas podem conter bastante ruído. O aumento do número de *beacons* utilizados na localização de um alvo aumenta a precisão

desta, porém, aumenta também a área de incerteza – onde poderá estar o alvo, a partir das interseções das circunferências descritas.



**Figura 2.2:** Triangulação com três âncoras, posicionadas em  $p_1$ ,  $p_2$  e  $p_3$ , e alvo localizado em  $p_0$  [5]

Formalmente, considerando  $N$  *beacons* utilizados ( $N \geq 3$ ), um alvo  $A$  e tendo estimado a distância  $d_N$  (do alvo ao *beacon*) recorrendo a um modelo de propagação, obtém-se um sistema de  $N$  equações quadráticas (2.3),

$$\begin{cases} d_1^2 = (x - x_1)^2 + (y - y_1)^2 \\ d_2^2 = (x - x_2)^2 + (y - y_2)^2 \\ \dots \\ d_N^2 = (x - x_N)^2 + (y - y_N)^2 \end{cases} \quad (2.3)$$

em que  $x_N$  e  $y_N$  correspondem às coordenadas de cada *beacon*,  $P_N(x_N, y_N)$ .

Idealmente, a interseção das  $N$  equações seria um único ponto, mas, porque na realidade não é o que acontece, o que se obtém é uma área de incerteza. É preciso encontrar então a posição mais provável do alvo,  $P_A$ , dentro da referida área. Esta é uma das razões pelas quais, muitas vezes, os algoritmos geométricos são complementados com algoritmos *Optimization-based* (2.4.1.B).

O algoritmo de Triangulação [30], por sua vez, utiliza medições de *AoA*, entre, no mínimo, um alvo e 2 âncoras, num cenário 2D, e 3 âncoras no caso 3D. Geometricamente, traça-se uma reta que é, na prática, um prolongamento do ângulo correspondente a cada âncora (dada uma certa origem de referência); a interseção das duas (ou mais) retas consiste na localização do alvo. Esta consegue alcançar-se com recurso a algumas relações trigonométricas, ou de forma mais simples, com um sistema de equações lineares.

### 2.4.1.B Optimization-Based

É inegável o facto de os algoritmos baseados em métodos geométricos serem bastantes intuitivos. Porém, é importante não esquecer que esses métodos apoiam-se em dados que podem ser (e geralmente são) erráticos, já que as observações feitas *indoor* são ruidosas e poderão conter, adicionalmente, erros (de medição, por exemplo).

Um dos algoritmos propostos em [28] foi o Linear Least Squares (LLSQ), que consiste em linearizar um sistema e resolvê-lo no sentido dos mínimos quadrados, o que se obtém subtraindo cada uma das  $N$  equações com as  $N-1$  restantes. Matricialmente, em  $\mathbb{R}^2$ , o sistema linear obtido pode ser descrito por (2.4)

$$A(P_A) = b \quad (2.4)$$

onde  $P_A$  é a posição desconhecida pertencente ao alvo e onde

$$A = \begin{bmatrix} x_1 - x_N & y_1 - y_N \\ x_2 - x_N & y_2 - y_N \\ \dots & \dots \\ x_{N-1} - x_N & y_{N-1} - y_N \end{bmatrix} \quad (2.5)$$

$$b = \frac{1}{2} \begin{bmatrix} x_1^2 - x_N^2 + y_1^2 - y_N^2 + d_1^2 + d_N^2 \\ x_2^2 - x_N^2 + y_2^2 - y_N^2 + d_2^2 + d_N^2 \\ \dots \\ x_{N-1}^2 - x_N^2 + y_{N-1}^2 - y_N^2 + d_{N-1}^2 + d_N^2 \end{bmatrix} \quad (2.6)$$

A solução encontrada é expressada por (2.7)

$$P_T = (A^T A)^{-1} A^T b \quad (2.7)$$

Outro estimador bastante conhecido e utilizado é o estimador Maximum Likelihood (ML). Ao contrário do que acontece com o estimador por LLSQ, o qual assume que a distribuição do ruído das medições efetuadas não é conhecida, o de ML assume-a conhecida e tem como objetivo maximizar a verosimilhança do vetor de parâmetros face às observações realizadas [31] [32]. Apesar de ser assintoticamente eficiente, este estimador pode ser altamente não linear e não convexo, o que torna difícil encontrar uma solução ótima. Para esse efeito, não poderão ser utilizados algoritmos iterativos, pois seria necessário que convergissem para um mínimo ou para um ponto de sela (*saddle point*). Para ultrapassar esta dificuldade, é comum recorrer-se a técnicas de relaxação convexas, estimadores lineares (como o LLSQ), algoritmos de *grid search*, entre outros [33].

## 2.4.2 Algoritmos Range-Free

Os sistemas que utilizam estas soluções continuam a precisar de efetuar medições (Secção 2.2), contudo estas não são utilizadas diretamente para inferir distâncias. O que acontece é que estas medições são comparadas com outras medições de referência (feitas *à priori*) e a partir daí é encontrada a localização. Por isto, estes sistemas precisam de uma configuração que poderá utilizar processos complexos, sendo demorada e dispendiosa.

### 2.4.2.A Fingerprinting

O método de *Fingerprinting* exige duas fases: *offline* e *online*. Na fase *offline* é feito o mapeamento do espaço: são coletadas amostras de RSS e associadas a coordenadas, formando o *radio map* de todo o espaço. Durante a fase *online*, o sistema compara as observações medidas com as constantes no *radio map*, para determinar qual a localização do alvo. Formalmente, definindo  $RSS_A$  como as observações de RSS correspondentes ao alvo A, num espaço S,  $F_l$  como a *fingerprint* do local b e c a função de custo (que tem como domínio todas as *fingerprints* possíveis para os *beacons* vizinhos), a posição do alvo,  $P_A$ , pode ser descrita por (2.8)

$$P_A = \arg \min c(RSS_A, F_l) \quad (2.8)$$

A minimização da função de custo  $c$  pode ser resolvida utilizando métodos probabilísticos, Support Vector Machines (SVM) [34], entre outros. Desta forma, verifica-se que esta técnica não relaciona as medições de RSS diretamente com as distâncias (entre *beacons* e alvo), mas sim com locais que foram introduzidos no mapeamento do espaço. Por isto, é pressuposto que as amostras de RSS associadas a um mesmo sítio não sofram alterações ao longo do tempo. Tal constitui um grande desafio, já que é sabido que os sinais são afetados pela difração, reflexão, *scattering*, entre outros fenômenos. Embora, aquando do mapeamento, uma maior granularidade possa permitir melhores resultados, não garante uma boa precisão.

# 3

## Metodologias

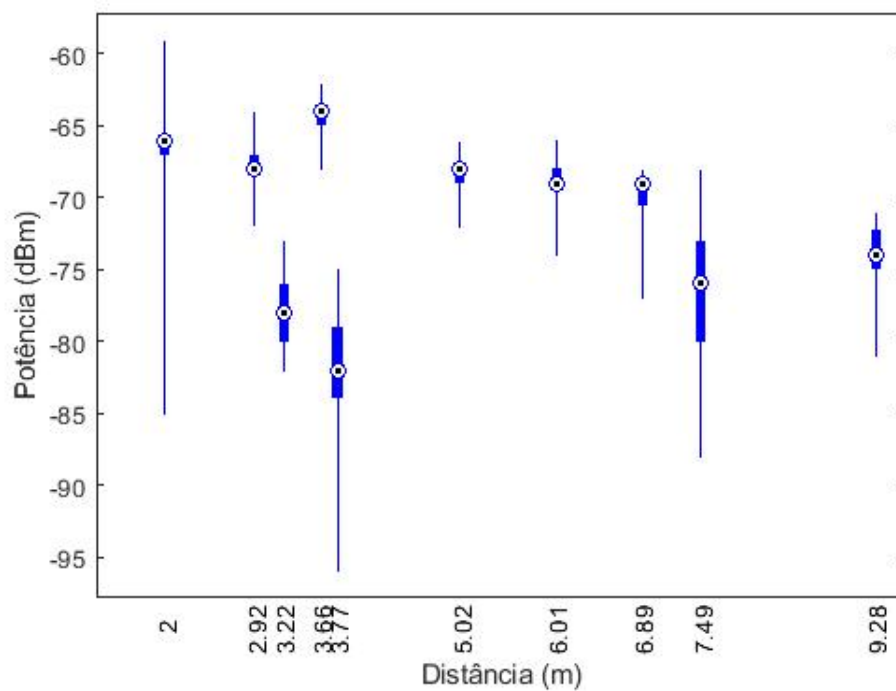
### Conteúdo

---

3.1 Multidimensional Scaling . . . . .	20
3.2 Análise Procrustes . . . . .	28

---

Nesta dissertação, constatou-se, como seria de esperar, que as medições de RSS são bastante ruidosas. Tal é apoiado pelo gráfico da Figura 3.1, que foi alcançado após se obterem os valores de RSS entre todos os pares de âncoras (que eram cinco, neste caso). Deveria ser possível identificar uma relação entre a distância entre âncoras e um valor de potência que, já que os valores se encontram na escala logarítmica, deveria ser linear. Descobriu-se então o mote de busca nesta dissertação: "Como conseguir localizações aproximadas quando a relação entre potências recebidas e distâncias é ténue e mal conhecida?"



**Figura 3.1:** Distância versus Potência.

Embora que com outros equipamentos e *setups*, na literatura existem já muitos trabalhos realizados no âmbito da localização recorrendo a RSS [35] [36] [37] [38]. Em vários deles se recorre a metodologias como LLSQ, estimadores de ML, relaxações convexas, entre outras. Na grande maioria destes trabalhos, os resultados em ambientes reais não foram os melhores, levando a crer que metodologias que envolvam cálculos sofisticados não são as ideais quando se pretende modelar dados com bastante ruído. Por isto, teve-se a intenção de se utilizar metodologias o mais simples e intuitivas possível. Por outras palavras, a abordagem tomada foi a de, a cada passo em frente, de entre as metodologias disponíveis escolher a de complexidade mais baixa que servisse o propósito proposto.



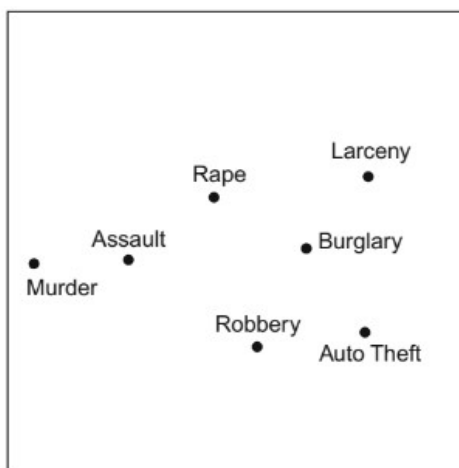
### 3.1 Multidimensional Scaling

O MDS surgiu na comunidade da Psicologia, sendo ainda hoje utilizado por psicólogos e sociólogos. Foi criado para ser utilizado como um modelo psicológico que explicasse, visualmente, como as pessoas formam opiniões e preferências por certas escolhas [6], bem como relações entre acontecimentos. Estas e outros objetos de estudo são representados no plano, como pontos.

Na tabela 4.16(a) podem ver-se as correlações entre a taxa de ocorrência de vários crimes que ocorreram em 50 estados dos EUA. Após escalar as correlações através do algoritmo MDS, obtém-se a sua representação visual (configuração de pontos), como se pode ver no gráfico da Figura 3.2. Neste, verifica-se que o crime 'Homicídio' ('*Murder*') está mais próximo do crime 'Assalto' ('*Assault*') do que do de 'Roubo' ('*Robbery*'), por exemplo, o que vai ao encontro dos dados da tabela 4.16(a), uma vez que a correlação entre 'Homicídio' e 'Assalto' é maior do que a entre 'Homicídio' e 'Roubo' ( $0.81 > 0.34$ ).

Crime	Murder	Rape	Robbery	Assault	Burglary	Larceny	Auto theft
Murder	1.00	0.52	0.34	0.81	0.28	0.06	0.11
Rape	0.52	1.00	0.55	0.70	0.68	0.60	0.44
Robbery	0.34	0.55	1.00	0.56	0.62	0.44	0.62
Assault	0.81	0.70	0.56	1.00	0.52	0.32	0.33
Burglary	0.28	0.68	0.62	0.52	1.00	0.80	0.70
Larceny	0.06	0.60	0.44	0.32	0.80	1.00	0.55
Auto theft	0.11	0.44	0.62	0.33	0.70	0.55	1.00

(a) Tabela de correlações



(b) Gráfico de proximidades

Figura 3.2: Tabela de correlações e gráfico correspondente [6]

No exemplo imediatamente acima, o input do algoritmo MDS, denominado de proximidades ou dis-

similaridades, foram as correlações. A designação de dissimilaridade/proximidade entre dois pontos (neste caso, os crimes) advém do facto de que quanto maior é o valor desta variável de entrada, menor será a distância entre os pontos, o que significa que estão mais próximos.

### 3.1.1 MDS Não-Métrico

MDS não métrico [6] ou *Non Metric MDS*, em inglês, também conhecido como *Ordinal MDS*, é o modelo de MDS mais conhecido e utilizado, dado o seu carácter flexível. A sua grande vantagem é a que admite como proximidades,  $\delta_{ij}$ , qualquer tipo de métrica ou valor derivado desta, desde que reflita uma relação entre os dados da amostra em estudo. A única condição imposta a esta métrica é a de que esta tem de ser ordinal, estar numa escala ordinal. Desta forma, as distâncias entre os pontos da configuração,  $X$ , obtida pelo MDS,  $d(X)_{ij}$ , estarão ordenadas de acordo com as proximidades  $\delta_{ij}$  ((3.1) e (3.2)). Apesar das distâncias  $d(X)_{ij}$  serem calculadas como distâncias euclidianas, existe sempre uma forte possibilidade destas não poderem ser interpretadas como distâncias propriamente ditas mas apenas como proximidade entre os pontos.

$$f : \delta_{ij} \rightarrow d_{ij}(X) \quad (3.1)$$

$$f : \delta_{ij} < \delta_{kl} \rightarrow d_{ij}(X) \leq d_{kl}(X) \quad (3.2)$$

Todas as outras variantes de MDS existentes apresentam os seus pontos positivos. Contudo, não foram aqui objetos de estudo, seja devido ao seu carácter mais restritivo em relação às proximidades, como acontece no MDS métrico, ou em relação à forma como estas foram obtidas, no caso do *Unfolding*, que assume que todas as proximidades são obtidas nas mesmíssimas condições. Apesar da utilização deste último modelo sair fora do âmbito desta dissertação, é interessante salientar que este nunca poderia ser utilizado neste cenário; é sabido que as *tags* (âncoras e *target*) se encontram em ambientes físicos reais e por variadas razões, como inclinação e direcção das *tags*, características dos equipamentos ou alterações no ambiente, é bastante provável que as condições de obtenção das medidas de RSS, no mesmo instante de tempo, variem de *tag* para *tag*.

O modelo de MDS escolhido para se obter a representação visual do *target* (Secção 4.3.4.D) foi o não métrico, pelas razões acima já enumeradas. Este é bastante parecido ao MDS Clássico (Secção 3.1.2), sendo que a principal diferença assenta na forma como se interpreta o resultado. Resumidamente, para se chegar desde as dissimilaridades  $\delta_{ij}$ , matriz  $\Delta$ , até às distâncias  $d_{ij}^1$ , é necessário resolver uma sequência de operações matriciais. Começa-se por centrar o a matriz  $\Delta^{(2)}$  (que contém os elementos  $\delta_{ij}^2$ ), que é uma matriz  $n \times n$  (em que  $n$  é o número de pontos), o que impede que a

---

<sup>1</sup>É de notar que, embora escritas de forma simplificada ( $d_{ij}$ ), estas distâncias referem-se sempre às distâncias entre os pontos da configuração  $X$ .

configuração "fuja" pelo plano e se afaste, obtendo-se  $B_\Delta$ , por (3.3). Nesta,  $J$  é obtida a partir de (3.4).

$$B_\Delta = -\frac{1}{2}J\Delta^{(2)}J \quad (3.3)$$

$$J = I - n^{-1}\mathbb{1}\mathbb{1}' \quad (3.4)$$

$$B_\Delta = Q\Lambda Q' \quad (3.5)$$

$$X = Q_+\Lambda_+^{\frac{1}{2}} \quad (3.6)$$

Posto isto, a configuração de pontos,  $X$  (matriz  $n \times m$  sendo  $m$  a dimensão do plano) é obtida através de (3.6), em que  $Q_+$  corresponde às primeiras  $m$  colunas de  $Q$  e  $\Lambda_+^{\frac{1}{2}}$  aos primeiros  $m$  valores próprios positivos da decomposição de  $B_\Delta$  (3.5). Se a matriz  $\Delta$  fosse preenchida por distâncias Euclidianas, todos os *eigenvalues* seriam positivos (ou negativos mas muito próximos de zero).

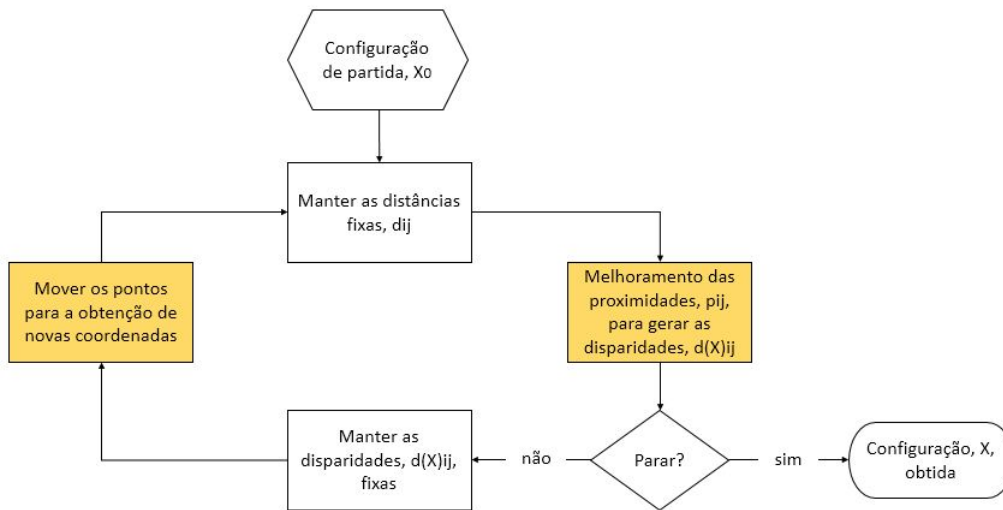
### 3.1.2 MDS Clássico vs MDS Iterativo

Existem dois métodos de se resolver um algoritmo MDS: na forma clássica e iterativamente.

A forma clássica assume que as proximidades são distâncias euclidianas e resume-se à resolução do algoritmo. Como se viu na subsecção anterior, tal não é mais do que uma série de operações algébricas, as quais assumem que a configuração  $X$  pode ser obtida através da decomposição da matriz  $B$  que, por sua vez, é obtida após se centrar duplamente (*double centering*) a matriz  $\Delta^{(2)}$ . Idealmente, as proximidades  $\delta_{ij}$  e as distâncias entre os pontos da configuração obtida,  $d_{ij}(X)$ , chamadas disparidades, seriam iguais. Contudo, quando se faz a diferença entre elas o que se obtém é uma matriz com valores diferentes de zero ( $\Delta - D$ ), excepto a diagonal cujos elementos serão sempre zero ( $(d(X))_{ii} = 0$ ). Dado o que as proximidades  $\delta_{ij}$  representam neste modelo, quanto maior for o módulo dos valores da matriz  $\Delta - D$ , mais longe estará a configuração obtida da realidade.

Quando os elementos da matriz  $\Delta$  são distâncias Euclidianas, a solução,  $X$ , obtida pelo MDS Clássico é já aquela que minimiza  $\Delta - D$ . Quando as dissimilaridades são outra métrica, é necessário utilizar o algoritmo MDS na forma iterativa. Tem-se como ponto de partida, uma configuração (aleatória ou não) e foca-se em minimizar uma função de custo que conduza à minimização de  $\Delta - D$ . No fundo, os pontos obtidos vão sendo, a cada iteração, movidos muito lentamente até ser obtido um valor de convergência da função de custo escolhida. Tal é esquematizado na Figura 3.3.

A função de custo mais conhecida quando se fala de qualquer modelo de MDS é o *Stress*. Este pode ser obtido pela expressão (3.7), onde  $w_{ij}$  são os pesos correspondentes a cada dissimilaridade  $\delta_{ij}$ . É um somatório residual de quadrados, positivo e quanto menor for o seu valor, mais aproximada da realidade está a configuração obtida (ou não, se este valor for um mínimo local e não global). Esta medida pode servir como '*fit index*', ou seja, serve para aferir se a configuração no final se aproxima da



**Figura 3.3:** Princípio de um algoritmo iterativo que utilize MDS.

realidade ou não, dadas as dissimilaridades fornecidas.

$$\sigma_r(X) = \sum_{i < j} w_{ij} (\delta_{ij} - d_{ij}(X))^2 \quad (3.7)$$

Existem vários algoritmos iterativos cujo objetivo é a minimização do *Stress*. Os considerados nesta dissertação seguem na secção seguinte.

### 3.1.3 SMACoF

Tal como sugerido na secção anterior, um algoritmo de minimização do *Stress*, pressupõe o uso de MDS de forma iterativa.

O Scaling by MAjorizing a COmplicated Function (SMACoF) [39] é um dos algoritmos mais utilizados, também conhecido como Majorization-Minimization (MM) [40], fazendo parte de vários programas de tratamento estatístico, como o Proxscal [41]. Como o seu nome sugere, este algoritmo consiste em minimizar o *Stress* através de majorização iterativa. A ideia principal, deste e de outros algoritmos da mesma família, é pegar numa função que seja (literalmente) complicada ( $f(x)$ ) e substituí-la por outra auxiliar que seja mais simples ( $g(x, z)$ ) e maior ou igual ( $f(x) \leq g(x, z)$ ), independentemente do número de variáveis, tal que  $z$  tem um valor fixo. Estes algoritmos têm uma menor probabilidade da solução encontrada ter ficado "presa" num mínimo local, pelo que este costuma ser global. Existem outras apreciações que se podem notar, como se verifica em [39] mas são apenas aqui enumeradas as consideradas relevantes para este trabalho.

Os passos seguidos pelo SMACoF são ilustrados no Algoritmo 3.1. Este tem como *input* uma configuração inicial,  $X_0$ , que pode ter sido gerada através do MDS clássico, aleatória ou outra fornecida

---

**Algoritmo 3.1: SMACoF**

---

**Dados:**  $X_0, \Delta$ **Resultado:** Configuração de pontos,  $X$  $k=0$ ; $\epsilon=1e-4$ ;calcular  $d_{ij}(X_0)$ ;aplicar algoritmo PAV em  $d_{ij}(X_0)$ :  $\hat{d}_{ij}$ ;normalizar  $\hat{d}_{ij}$ ;calcular o Stress:  $\sigma_r^{(0)} = \sigma_r(\hat{d}, X^{(0)})$ ; $\sigma_r^{(-1)} = \sigma_r^{(0)}$ ;**while**  $k=0$  ou  $(\sigma^{(k-1)} - \sigma^{(k)}) > \epsilon$  e  $k \leq$  máximo de iterações **do**    calcular a transformada de Guttman:  $X^k = n^{-1}B(Z)Z$ ;    calcular  $d(X^{(k)})_{ij}$ ;    aplicar algoritmo PAV em  $d_{ij}(X)$ :  $\hat{d}_{ij}$ ;     $\delta_{ij} = \hat{d}_{ij}$ ;    normalizar  $\hat{d}_{ij}$ ;    calcular o Stress:  $\sigma_r(\hat{d}, X^{(k)})$ ;     $Z = X^{(k)}$ ;**end**

---

pelo utilizador e também as dissimilaridades  $\delta_{ij}$ ,  $\Delta$ , que conduziram à tal configuração. As distâncias  $d_{ij}(X_0)$  tratam-se das distâncias euclidianas entre os pontos de  $X_0$ . Outra forma de pensar, será denotar as distâncias  $d_{ij}(X_0)$  como uma transformação das dissimilaridades  $\delta_{ij}$ , como em (3.8). Assim, é de esperar que  $d_{ij}(X_0)$  tenham "monotonia fraca" ("*weak monotonicity*") em ordem a  $\delta_{ij}$ , isto é, que as distâncias  $d_{ij}(X_0)$  sejam monotonicamente não decrescentes em relação a  $\delta_{ij}$ , (3.9).

$$d_{ij}(X_0) = f(\delta_{ij}) \quad (3.8)$$

$$\text{se } \delta_{ij} < \delta_{kl} \text{ então } d_{ij}(X) < d_{kl}(X) \quad (3.9)$$

Para garantir esta última relação e posteriormente se obterem as disparidades,  $\hat{d}(X)$ , recorre-se a um Pool-Adjacent-Violators Algorithm (PAVA). Este algoritmo "varre", a cada iteração, as distâncias  $d(X)$  e verifica se a condição (3.9) é cumprida. Se isso não acontecer, atribuem-se outros valores às disparidades  $\hat{d}_{ij}(X)$  e  $\hat{d}_{kl}(X)$  (que serão as distâncias  $d_{ij}(X)$  e  $d_{kl}(X)$ , respetivamente, na próxima iteração do PAVA), com vista a minimizar o número de iterações necessário. Neste trabalho, seguiu-se a proposta de [42] e atribuiu-se às disparidades a média das distâncias como em (3.12). Outra situação que (frequentemente) pode ocorrer é a existência de dissimilaridades iguais ( $\delta_{ij} = \delta_{kl}$ ) e é necessário "desempatar" as distâncias ("*break ties*"), decidir que valor as disparidades podem tomar nesta situação. Existem duas abordagens que se podem seguir: primária ("*primary approach*") ou secundária ("*secondary approach*"). Com a primeira, não existem nenhuma restrição, ou seja, as disparidades poderão

tomar qualquer valor ou serem iguais às distâncias

$$\text{se } \delta_{ij} = \delta_{kl} \text{ então } \hat{d}_{ij}(X) \in \mathbb{R} \text{ e } \hat{d}_{kl}(X) \in \mathbb{R} \quad (3.10)$$

Com a abordagem secundária, é obrigatório que as disparidades sejam iguais a um certo valor  $l$ ,

$$\text{se } \delta_{ij} = \delta_{kl} \text{ então } \hat{d}_{ij}(X) = \hat{d}_{kl}(X) = l, \quad l \in \mathbb{R} \quad (3.11)$$

A abordagem utilizada no SMACoF é a "primária", pois esta confere mais liberdade aos dados<sup>2</sup>, e nestes casos as disparidades tomam valores iguais aos das distâncias.

$$\delta_{ij} < \delta_{kl} \longrightarrow \begin{cases} \text{se } d_{ij}(X) \leq d_{kl}(X) & \text{então } \hat{d}_{ij} = d_{ij}(X) \text{ e } \hat{d}_{kl} = d_{kl}(X) \\ \text{se } d_{ij}(X) > d_{kl}(X) & \text{então } \hat{d}_{ij} = \hat{d}_{kl} = \frac{1}{2}(d_{ij}(X) + d_{kl}(X)) \end{cases} \quad (3.12)$$

Posto isto, normalizam-se as disparidades  $\hat{d}_{ij}$ , que são um vetor  $s \times 1$  no qual  $s = n(n-1)/2$ , para um valor fixo. Como se verá muito brevemente, é conveniente que este valor seja igual a  $s$ ,  $\eta_d^2 = n(n-1)/2$ .

Desdobrando a expressão (3.7), obtém-se a (3.19), que pode também ser representada por (3.20). Tendo-se normalizado as disparidades  $\hat{d}_{ij}$ , o termo  $\eta^2(X)$  é sempre constante, o que é uma simplificação ao cálculo de  $\sigma_r(X)$ .

Se a matriz  $W$ , cujos elementos são  $w_{ij}$ , for reduzível, então é possível se analisar os termos  $\eta^2(X)$ ,  $\rho(X)$  par a par. O termo  $\eta^2(X)$  é um somatório pesado dos quadrados das distâncias  $d_{ij}$ ; tenha-se como exemplo o par  $i = 1$  e  $j = 3$ , ter-se-á

$$d_{13}^2(X) = \sum_{a=1}^m x'_a (e_1 - e_3)(e_1 - e_3)' x_a = \sum_{a=1}^m x'_a \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} x_a = \sum_{a=1}^m x'_a A_{13} x_a \quad (3.13)$$

em que  $x_a$  é a coluna  $a$  da matriz  $X$ ,  $e_1$  a primeira coluna da matriz identidade ( $e'_1 = [1 \ 0 \ 0 \ 0]$ ) e  $e_3$  a terceira ( $e'_3 = [0 \ 0 \ 1 \ 0]$ ), sendo  $(e_1 - e_3)' = [1 \ 0 \ -1 \ 0]$ . Generalizando para todos os pares  $ij$  e somando ter-se-á

$$\eta^2(X) = \sum_{i < j} w_{ij} d_{ij}^2 = \text{tr} X' \left( \sum_{i < j} w_{ij} A_{ij} \right) X = \text{tr} X' V X \quad (3.14)$$

sendo  $V$  uma matriz na qual  $v_{ij} = -w_{ij}$  para  $i \neq j$  e  $v_{ii} = \sum_{j=1, j \neq i}^n w_{ij}$  para os elementos da sua diagonal. O termo  $-2\rho(X)$  é também um somatório pesado, mas das distâncias  $d_{ij}$  "planas". Utilizando

<sup>2</sup>É importante não esquecer que as proximidades/dissimilaridades utilizadas na maioria dos problemas utilizando MDS não se tratam de distâncias euclidianas, mas sim de correlações ou outras métricas e que geralmente possuem erros. Desta forma, com quanto mais liberdade se poder efetuar o tratamento dos dados, melhores resultados se obterão.

a desigualdade de Cauchy-Schwarz substituindo algumas variáveis obtém-se

$$\sum_{a=1}^m (x_{ia} - x_{ja})(z_{ia} - z_{ja}) \leq \left( \sum_{a=1}^m (x_{ia} - x_{ja})^2 \right)^{\frac{1}{2}} \left( \sum_{a=1}^m (z_{ia} - z_{ja})^2 \right)^{\frac{1}{2}} = d_{ij}(X) d_{ij}(Z) \quad (3.15)$$

e

$$-d_{ij}(X) = -\frac{\sum_{a=1}^m (x_{ia} - x_{ja})(z_{ia} - z_{ja})}{d_{ij}(Z)} \quad (3.16)$$

podendo-se considerar

$$\sum_{a=1}^m (x_{ia} - x_{ja})(z_{ia} - z_{ja}) = \text{tr} X' A_{ij} Z = \quad (3.17)$$

e chegando a

$$-\rho(X) = -\sum_{i<j} (w_{ij} \delta_{ij}) d_{ij}(X) \leq -\text{tr} X' \left( \sum_{i<j} b_{ij} A_{ij} \right) Z = \text{tr} X' B(X) Z \quad (3.18)$$

sendo B alcançado por (3.21).

$$\sigma_r(X) = \sum_{i<j} w_{ij} \delta_{ij}^2 + \sum_{i<j} w_{ij} d_{ij}^2(X) - 2 \sum_{i<j} w_{ij} \delta_{ij} d_{ij}(X) \quad (3.19)$$

$$\sigma_r(X) = \eta_d^2 + \eta^2(X) - 2\rho(X) \quad (3.20)$$

$$B(Z) : b_{ij} = \begin{cases} -\frac{w_{ij} \delta_{ij}}{d_{ij}}, i \neq j \text{ e } d_{ij}(Z) \neq 0 \\ 0, i \neq j \text{ e } d_{ij}(Z) = 0 \end{cases} \quad b_{ii} = -\sum_{j=1, j \neq i}^n b_{ij} \quad (3.21)$$

Após se terem todas as variáveis bem definidas, entra-se no ciclo ("loop") de iterações, que continua sem parar até que o número máximo de iterações, definido pelo utilizador, seja alcançado ou até que os valores de Stress convirjam para um determinado valor. Nesta situação, entenda-se por convergência a diferença entre o *Stress* calculado na iteração atual e o da iteração anterior ser menor que um determinado valor, geralmente de ordem  $10^{-4}$  ( $\sigma^{(k-1)} - \sigma^{(k)} < \epsilon$ ).

A cada iteração é necessário atualizar a nova configuração de pontos  $X^k$ , recorrendo-se assim à transformada de Guttman, em (3.22). Se todos os pesos  $w_{ij}$  forem igual a 1, então  $V^+ = n^{-1} J$  e simplifica-se  $X^k$  (3.23), obtendo-se  $J$  com a expressão (3.4). Calculam-se as distâncias  $d_{ij}(X^k)$ , aplica-se o PAVA e antes de as normalizar igualam-se as dissimilaridades a estas (que irão servir como dissimilaridades no PAVA da próxima iteração). Finalmente, calcula-se o *Stress* (3.20).

Quando o SMACoF termina, é suposto que a configuração de pontos obtida corresponda ao mínimo global da função de *Stress*.

$$X^k = V^+ B(Z) Z \quad (3.22)$$

$$X^k = n^{-1}B(Z)Z \quad (3.23)$$

### 3.1.4 MDSCALE - MATLAB

Para além do algoritmo SMACoF, neste trabalho fez-se também uso da função *mdscale* [43] fornecida pelo MATLAB. Nesta função é implementado um algoritmo, que utiliza MDS, tal como o SMACoF. Estes dois algoritmos são bastante parecidos, estando a grande diferença no processo de atualização da configuração de pontos. Para efeitos de simplificação da escrita, o algoritmo presente na função *mdscale* será denominado com o mesmo nome.

Seguindo a sequência do Algoritmo 3.1, a primeira diferença é que no MDSCALE primeiro normalizam-se as distâncias  $d_{ij}(X)$  e só depois é aplicado o PAVA, que tem como saída as disparidades  $\hat{d}_{ij}$ . Diferentemente do algoritmo anterior, não existe a necessidade de normalizar as distâncias para um valor fixo, levando a que estas sejam apenas multiplicadas por um escalar,  $a$ , cuja fórmula segue em (3.24). Também aqui se tem a "abordagem primária" no que toca a "empates" entre dissimilaridades: quando iguais, as distâncias podem tomar qualquer valor.

$$a = \frac{\max(\text{dissimilaridades } \delta_{ij})}{\max(\text{distâncias } d_{ij})} \quad (3.24)$$

Quando as distâncias não obedecem ao princípio da "monotonia fraca", o valor a tomar por estas é o mostrado em (3.25).

$$\delta_{ij} < \delta_{kl} \longrightarrow \begin{cases} \text{se } d_{ij}(X) \leq d(X)_{kl} & \text{então } \hat{d}_{ij} = d_{ij}(X) \text{ e } \hat{d}_{kl} = d_{kl}(X) \\ \text{se } d_{ij}(X) > d(X)_{kl} & \text{então } \hat{d}_{ij} = \hat{d}_{kl} = \text{média pesada de } d(X) \end{cases} \quad (3.25)$$

De seguida é calculado o *Stress*, através de 3.7, e seu respetivo gradiente, em ordem a  $X$ , mantendo as disparidades  $\hat{d}_{ij}$  constantes. A minimização do *fit index* é feita através de um método de *line search*, o método de "Polak-Riviere" [44], que encontra a direção de procura do Gradiente Conjugado (CG) (*Conjugate Gradient Search Direction*). Na literatura, é comum encontrar a seguinte afirmação "Os algoritmos de CG estão entre os *steepest descent* e o Método de Newton", isto porque eles convergem mais rapidamente que o primeiro e têm um custo computacional menor que o segundo. Formalmente, o método escolhido (tal como os outros algoritmos de CG) tem como objetivo minimizar uma função, pelo menos quadrática, na forma (3.26), sob a condição que a sua solução  $x^*$  é única e também é a solução da equação linear (3.27), sendo alcançada  $k$  iterações depois do início do ciclo.

$$f(x) = \frac{1}{2}x'Qx - b'x, Q \in \mathbb{R}^{n \times n} \quad (3.26)$$



$$Qx = b \quad (3.27)$$

Tomando como *starting point*  $x_0$ , o qual, para o MDSCALE, será a configuração  $X$  inicial, e como direção de procura inicial,  $d_0$ , o simétrico do gradiente de  $f$ ,  $g_0$ , tal que

$$d_0 = -g_0 = b - Qx_0 \quad (3.28)$$

iniciam-se as iterações que têm como *update rule* (termo de atualização da configuração)  $\alpha_k$

$$\alpha_k = \frac{g'_k g_k}{d'_k Q d_k} \quad (3.29)$$

atualizando a solução  $x_{k+1}$ , o gradiente  $g_k$  e a direção (do mínimo) de procura  $d_{k+1}$

$$\begin{cases} x_{k+1} = x_k + \alpha_k d_k \\ g_k = Qx_k - b \\ d_{k+1} = -g_{k+1} + \beta_k d_k \end{cases} \quad (3.30)$$

em que  $\beta_k$  é

$$\beta_k = \frac{(g_{k+1} - g_k)}{g'_k g_k} \quad (3.31)$$

Resumidamente o MDSCALE encontra um *upper bound*, majorizante, onde o *Stress* é maior, encontrando o máximo *step length*, que essencialmente é o espaço de procura, procurando o mínimo entre zero e esse valor através de *line search*. Poderá acontecer que o *Stress* da iteração anterior seja menor que o da atual; isso significa que primeiramente o *Stress* esteve no seu mínimo global, tendo-se desviado para um mínimo local. No caso de tal acontecer, o algoritmo termina, sem solução possível.

## 3.2 Análise Procrustes

Segundo a mitologia grega, Procrusto era um criminoso que obrigava viajantes a caber de forma exata na sua cama, quer fosse esticando os seus membros até à exaustão do mesmos ou cortando-os. Apesar de bastante macabro, é uma analogia perfeita ao objetivo da Análise Procrustes (*Procrustes Analysis*) [45]. Dada uma configuração de pontos,  $Y$ , pretende-se "fazê-la caber" numa configuração  $X$ , isto é, que os pontos de  $Y$  ocupem as mesmas posições dos de  $X$ . Tal é conseguido através de transformações rígidas: rotação, reflexão, translação e dilatação. A este processo dá-se o nome de Análise Procrustes.

As transformações rígidas, lineares, são encontradas através da minimização do somatório dos erros quadrados, Sum Squared Error (SSE),  $L(T)$  (3.32) em que  $T$  é a matriz ótima de rotação e

reflexão.

$$\begin{aligned}
 L(T) &= \|X - YT\|^2 \\
 &= \text{tr}(X - YT)'(X - YT) \\
 &= \text{tr}(X - YT)(X - YT)' \\
 &= \text{tr}X'X + \text{tr}T'Y'YT - 2\text{tr}X'YT \\
 &= \text{tr}X'X + \text{tr}Y'Y - 2\text{tr}X'YT
 \end{aligned} \tag{3.32}$$

Como os termos  $\text{tr}X'X$  e  $\text{tr}Y'Y$  não dependem de  $T$ , pode-se escrever  $L(T)$  na forma

$$L(T) = c - 2\text{tr}X'YT \tag{3.33}$$

Posto isto, segue-se uma série de operações algébricas que levam à possibilidade de se efetuarem os seguintes passos:

1. Cálculo da matriz  $C$

$$C = X'JY \tag{3.34}$$

2. Decomposição Singular Value Decomposition (SVD) de  $C$

$$C = P\phi Q' \tag{3.35}$$

3. Cálculo da matriz ótima de rotação,  $T$

$$T = QP' \tag{3.36}$$

4. Se o determinante de  $T$  for negativo ( $\det T < 0$ ) é um aviso de que é necessário fazer uma reflexão. Para isso iguala-se a matriz  $Q$  à sua simétrica,  $Q = -Q$ , e recalcula-se  $T$ ;

5. Cálculo do factor de dilatação (*scaling factor*)  $s$

$$s = \frac{\text{tr}X'JYT}{\text{tr}Y'JY} \tag{3.37}$$

6. Cálculo do vetor de translação ótimo  $t$

$$t = n^{-1}(X - sYT)'1 \tag{3.38}$$

7. Cálculo da configuração obtida  $Z$

$$Z = sYT + t \tag{3.39}$$

A matriz  $J$  nas equações (3.34) e (3.37) é a referida na (3.4). A matriz  $Z$  é a melhor aproximação possível (utilizando esta análise) dos pontos de  $Y$  aos pontos de  $X$ .

# 4

## Resultados

### Conteúdo

---

4.1 Trabalho Experimental . . . . .	33
4.2 Resultados Preliminares . . . . .	36
4.3 Resultados . . . . .	42

---



## 4.1 Trabalho Experimental

### 4.1.1 Sensores (Tags)

Os sensores utilizados no trabalho experimental desta dissertação, até aqui chamados mais comumente por *tags*, foram os *Smart Proximity Beacons* desenvolvidos pelo *Open Beacon*, que contêm o chip nRF51822, pertencente à série de chips System On a Chip (SoC) da Nordic Semiconductors [46]. No projeto original da Thales, a empresa utilizou uma placa impressa com os mesmos componentes da tag *Open Beacon* e mais alguns, como diferentes Light Emitting Diodes (LEDs), necessários à sua atividade experimental 4.1.2. É importante salientar que ambos os sensores têm um antena 2.4 GHz compatível com BLE.

Mais detalhes acerca do hardware das tags *Open Beacon* podem ser consultados na sua página [2].



Figura 4.1: Sensores utilizados. [3]

### 4.1.2 Experiência Thales

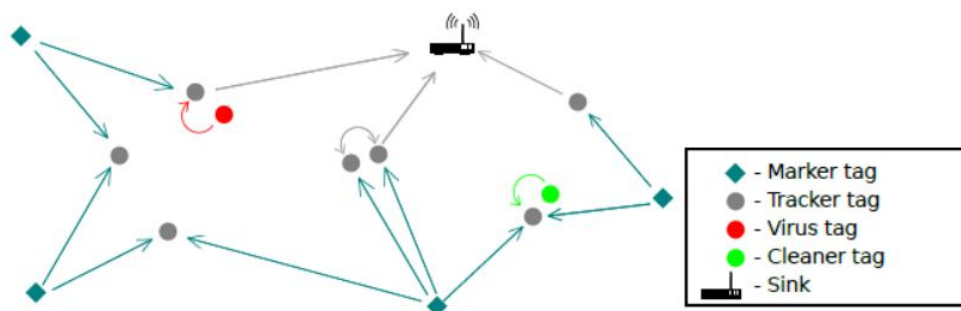
A experiência desenvolvida pela Thales tinha como objetivo recriar um cenário de propagação de um vírus na comunidade. Para isso, existiam quatro tipos de *tags*: *markers*, *trackers*, *virus* e *cleaners*, para além do *sink node* (fig.4.2) que atuava como estação-base (BS) para as *trackers* (cada BS tem um alcance até aproximadamente 100 m [47]). As primeiras estavam dispostas em sítios estratégicos e imóveis (na sala principal de exposições temporárias do pavilhão Ciência Viva), atuando como âncoras e as *trackers* e *virus* eram carregadas por pessoas que andavam livremente, comportando-se como *targets*. Quando uma *tag virus* se aproximava de uma *tracker* não infetada havia grande chance de a infetar e quando uma *tracker* infetada se aproximava poucos centímetros de uma *cleaner* (que se encontrava imóvel na sala) havia a (baixa) possibilidade de esta "curar" a *tracker* e de a tornar imune, dependendo de quem recebia primeiro um pacote<sup>1</sup>.

<sup>1</sup>A comunicação entre *tags* é feita por pacotes, recorrendo a sinais RF. O protocolo específico não foi abordado nesta dissertação, pois não se prende com o seu objetivo.



**Figura 4.2:** OpenBeacon estação-base (EasyReader).

Na Figura 4.3 está esquematizada a comunicação entre *tags*. Este esquema de comunicação é não-cooperativo: as âncoras não comunicam entre si nem com a BS mas apenas com os *targets*. Por sua vez, estes últimos comunicam entre si e com os outros elementos da rede.



**Figura 4.3:** Esquema de comunicação entre tags [3].

#### 4.1.2.A Comunicação entre tags

A tabela presente na Figura 4.4 lista a troca de pacotes entre os componentes da rede. Como esperado num cenário não cooperativo, o *Sink* apenas recebe os pacotes *tracker* enviados pelas *trackers*; estes pacotes contêm a medição de potência entre a própria *tag* e uma *marker*. As *tags* que atuam como *targets* escutam e enviam pacotes de proximidade, os quais não contêm valores de RSS medidos (o que não é necessário) mas apenas um indicador em como a respetiva *tag* está próxima, para a "infectar" ou "limpar". Há que notar que as *tags virus* e *cleaner* transmitem os seus pacotes (de proximidade) com a menor potência, uma vez que se pressupõe que devem ser recebidos por outra *tag* quando esta se encontra muito próxima.

Node	Sends Proximity	Sends Tracker	Listens for Proximity	Listens for Tracker	TX Power
Sink	X	X	X	✓	N/A
Marker tags	✓	X	X	X	4 dBm
Tracker tags	✓	✓	✓	X	-16 dBm
Virus tags	✓	X	✓	X	-30 dBm
Cleaner tags	✓	X	✓	X	-30 dBm

**Figura 4.4:** Troca de pacotes entre tags e BS [3].

Na tese [3] o autor fez uma análise do tempo entre pacotes. Estimou-se que, em média, a cada 25 ms é enviado um pacote de proximidade, pelas *tags* correspondentes e que após o décimo envio ( $\approx 250ms$ ) existe uma janela de recepção de 5 ms. Por outro canal de comunicação, é feito a cada segundo o envio de um pacote *tracker*.

### 4.1.3 Montagens

No contexto deste trabalho, chamou-se "montagem" ao conjunto das etapas que compõem o processo de obtenção das medições entre tags, desde a implementação física das âncoras à obtenção das medições, recorrendo ao *firmware* disponibilizado pela Thales<sup>2</sup>.

#### 4.1.3.A Cenário Cooperativo

Em todas as montagens realizadas se teve uma abordagem cooperativa. Todas as tags comunicaram entre si, enviando (todas) pacotes *tracker* (4.1.2.A) à BS. A potência de transmissão foi fixada nos 4 dBm, como tentativa de melhorar a performance, quer isto dizer, como tentativa do sinal ser detetado a uma maior distância.

Foi utilizado apenas um *target*, a única *tag* móvel, por questões relacionadas com equipamento. De notar que a metodologia baseada em MDS acomoda o cenário cooperativo, pois determina várias posições relativas em simultâneo, levando a que não se trate de um trabalho "puramente" *single-source*.

#### 4.1.3.B Disposição espacial e ambientes

Durante o decorrer da fase de trabalho experimental, foram realizadas várias montagens, em ambientes diferentes e com as âncoras dispostas de maneiras diferentes. Tal abordagem teve o intuito de ver como diferentes aspectos se refletiriam no resultado final. Os vários aspetos tidos em conta foram:

- altura, LoS e distância entre tags;

<sup>2</sup>Este firmware deriva do disponibilizado pelo Open Beacon.



- "demografia" do espaço (se existiam pessoas entre o perímetro da configuração espacial formada pelas âncoras);
- predominância de materiais no ambiente (alguns ambientes tinham muitas estruturas metálicas, por exemplo)
- sinais que pudessem causar interferência com os de transmissão das *tags*.

Os efeitos da inclinação dos sensores não foram objetos de estudo, uma vez que se sabia *a priori* que as antenas dos mesmos são altamente diretivas. Sabendo isto, em todas as ocasiões os sensores foram mantidos em tal posição que garantisse a sua antena na vertical (direção de propagação na horizontal, ortogonal à antena) - mais detalhes na Secção 4.2.1.

#### 4.1.3.C Obtenção das medições de RSS

Através do firmware disponibilizado pela Thales, que tem por base o de *open source* encontrado em [48], é possível extrair os dados que a BS recebe das *tags*. É assim que se conseguem obter as medições entre pares de sensores. Para além do valor de potência entre um par, o *output* do programa também informa qual a *tag* que transmitiu o sinal (*sender*) e qual foi a recetora (*receiver*).

## 4.2 Resultados Preliminares

A primeira subsecção desta secção descreve sumariamente o comportamento das antenas dos sensores, através de uma *task force* realizada em conjunto com o Vasco Rato [3]. Nas restantes subsecções é ilustrada a "jornada" percorrida no tratamento dos dados obtidos (medições). Foram tomadas várias decisões sempre tendo em mente o resultado final, com vista a melhorá-lo, mas também com uma forte vertente de experimentação e curiosidade. É de recordar que o resultado final deverá ser a representação visual da trajetória percorrida pelo *target*.

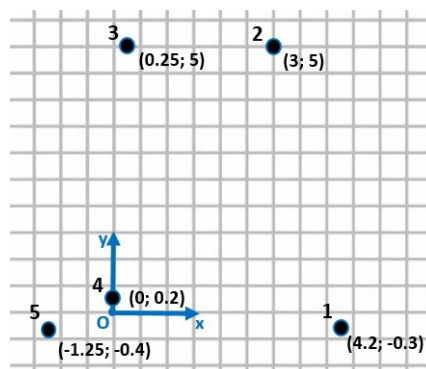
Nas subsecções seguintes, os gráficos que serão apresentados referem-se à montagem "j01"<sup>3</sup>. O espaço físico onde a montagem foi realizada foi a "sala do tanque" no laboratório do ISR [49] (no *Campus* do TagusPark). Como se pode ver na Figura 4.5, esta sala contém muitos vidros e algumas estruturas metálicas; ambos os materiais são refletores, sobretudo o metal (o que gera fenómenos de *multipath* e *fading*). Nesta montagem as *tags* estão todas sensivelmente à mesma altura e em LoS. O único momento em que isso pode não acontecer é quando o *target* se interpõe entre *tags*, o que pode causar forte atenuação do sinal, visto que a antena é carregada por uma pessoa e a água contida no corpo humano é um forte atenuante.

<sup>3</sup>Durante a fase de trabalho experimental foi criado uma espécie de "diário de bordo". Neste documento foi dado a cada experiência um nome em que a letra representa o espaço físico e os números a ordem em que esta foi realizada.



**Figura 4.5:** Disposição das âncoras - DSOR.

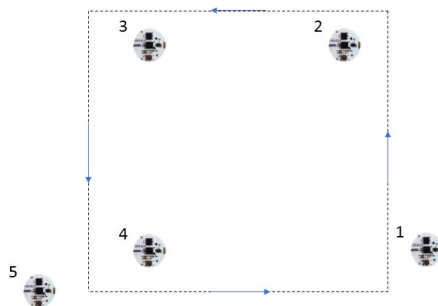
As coordenadas ocupadas pelas âncoras podem ser vistas na Figura 4.6, considerando um espaço virtual ao redor da configuração de âncoras de 5.4 m x 5.45 m. Poderiam ser tomadas as dimensões de toda a sala e como origem um dos cantos da sala, contudo isso não traria nenhum benefício e tornaria as medições de distância<sup>4</sup> mais difíceis, o que poderia aumentar os erros de medição.



**Figura 4.6:** Coordenadas das âncoras, com origem na tag 4.

O *target* descreveu um quadrado em volta das âncoras, durante cerca de 30 segundos. A trajetória pode ser visualizada na Figura 4.7.

<sup>4</sup>As medições entre *tags* e outras necessárias foram efetuadas com uma régua laser.

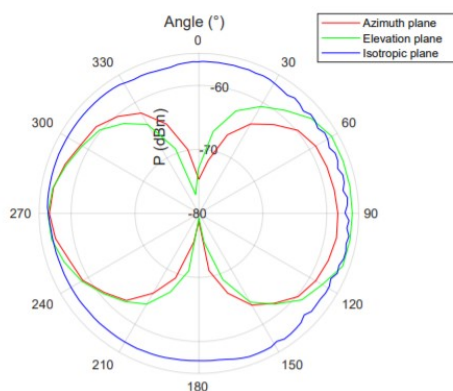


**Figura 4.7:** Trajetória descrita pelo *target*, entre as âncoras (de 1 a 5).

### 4.2.1 Orientação dos sensores

Após analisar alguns ficheiros disponibilizados pela Thales, encontrou-se a informação, dada por Milosch Meriac<sup>5</sup>, de que as antenas dos sensores seriam diretivas e estes deveriam ser posicionados de tal forma que a sua antena ficasse na vertical. De forma a corroborar esta tese, foi realizada uma experiência na câmara anecóica do *campus* da Alameda, descrita com detalhe na dissertação "RSS-based Indoor Localization using a Network of Open-Source Wireless Tags" [3].

Verificou-se, então, que a antena se comporta como um dipolo magnético, cuja direção de ganho máximo é a ortogonal ao plano da antena, como se vê na Figura 4.8. Por isto, idealmente os sensores devem ser posicionados à mesma altura e com a antena "voltada" para cima.



**Figura 4.8:** Diagrama de propagação da antena de um sensor (retirada de [3]).

### 4.2.2 Potência vs Distância

Atentando na Equação (2.1), facilmente se infere que se ambas as variáveis estiverem na escala logarítmica, a sua relação é linear e decrescente

<sup>5</sup>Fundador do OpenBeacon.

$$P_{ij} = a - \gamma d_{ij} \quad (4.1)$$

em que  $P_{ij}$  é o valor de RSS entre as *tags*  $i$  e  $j$  e  $d_{ij}$  a distância entre as mesmas.

Como seria excessivo mostrar um histograma por cada par de âncoras, criou-se o gráfico (*boxplot* do MATLAB) da Figura 3.1. Neste, os pontos pretos ("olhos") representam a mediana das amostras, o retângulo azul os valores dentro dos quartis de 25% e 75% e os traçados os restantes valores (não *outliers*). Aqui, repara-se que a relação entre a(s) distância(s) e a(s) potência(s) não é linear, de facto, é até quase impossível encontrar qualquer relação entre as mesmas.

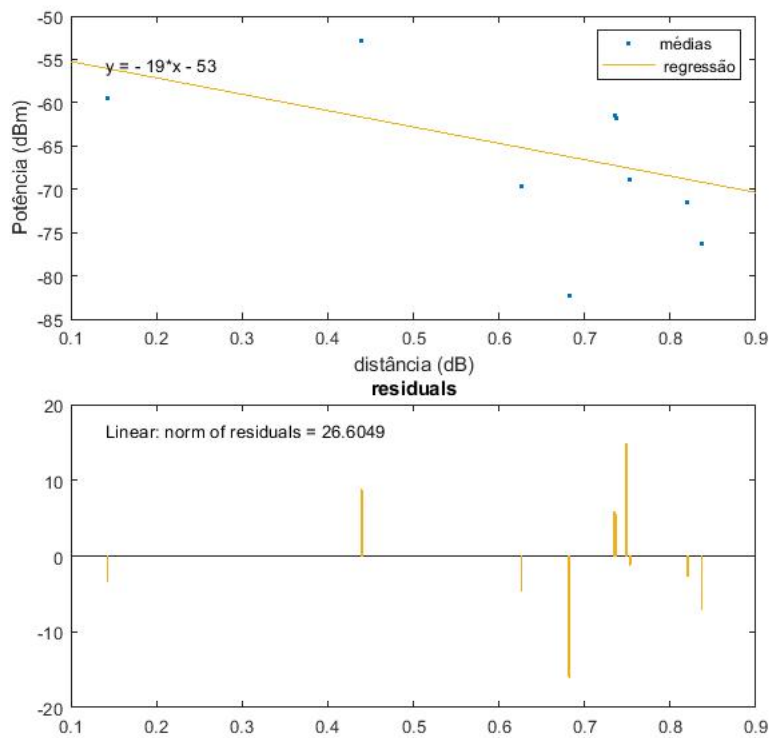
### 4.2.3 Preparação das potências

A ferramenta fundamental utilizada neste trabalho para atacar o problema é o algoritmo MDS. Decidiu-se desde o início que a entrada desse algoritmo, a matriz de dissimilaridades  $\Delta$ , seria medições de RSS, valores de potência. Para além disso, decidiu-se que se iria obter uma configuração de pontos a cada segundo, visto que, como referido na Secção 4.1.2.A, as *tags* transmitem as suas medições de RSS a cada segundo. Porém, esta estimativa de transmissão de um pacote por segundo é um valor médio e por isso, em adição à possível ocorrência de erros de colisão e/ou pacotes corrompidos, podem aparecer algumas lacunas de valores.

#### 4.2.3.A Regressão Linear

Tentando sempre utilizar as metodologias disponíveis mais simples, pensou-se numa simples regressão linear que servisse para aproximar os valores de potências, de forma a que tal aproximação correspondesse a valores de potência menos ruidosos.

Tomando as médias dos valores de RSS trocados entre cada par (utilizando as medianas seria semelhante), representou-se a reta de regressão linear que mais se aproximava de todos os dados, obtendo-se os gráficos da Figura 4.9.



**Figura 4.9:** Em cima: reta de regressão linear. Em baixo: representação dos resíduos da reta de regressão.

Apesar de se ter conseguido encontrar uma reta de regressão linear não horizontal (pior cenário de todos) e esta ser decrescente, como se pode observar no gráfico de baixo, alguns resíduos chegam a ter uma norma de cerca de 15 dB, o que não é de todo favorável.

Um indicador que mostra se o modelo (os pontos dos gráfico) têm uma componente de erro aleatório pequena é o SSE

$$SSE = \sum_{i=1}^n w_i (y_i - f_i)^2 \quad (4.2)$$

em que  $w_i$  é o peso aplicado a cada medição  $y_i$  (neste trabalho considerou-se que todas as medições teriam sempre o mesmo peso e este seria igual a 1) e  $f_i$  representa o ponto da reta de regressão correspondente a  $y_i$ . Obteve-se um valor de SSE de 707.8 o que é péssimo, pois este deveria estar próximo de 0. Da mesma forma, a métrica estatística Root Mean Square Error (RMSE) [50] também deveria de aproximar de zero mas toma o valor de 9.4. O RMSE indica se a reta de regressão representa suficientemente o desvio padrão da componente aleatória dos dados (4.4).

$$MSE = \frac{SSE}{n^{\circ} \text{ de dados da amostra}} \quad (4.3)$$

$$RMSE = \sqrt{MSE} \quad (4.4)$$

O indicador R-Square (4.5) também está aquém do desejado, sendo 0.1686 quando deveriam tomar um valor próximo de 1. Este indicador serve para medir se a reta de regressão encontrada representa bem a variação da amostra.

$$R - Square = 1 - \frac{\sum_{i=1}^n w_i (y_i - f_i)^2}{\sum_{i=1}^n w_i (y_i - y_{av})^2} = 1 - \frac{SSE}{SST} \quad (4.5)$$

Como se pode verificar por análise da Tabela 4.1 todos as métricas estatísticas utilizadas indicam que a reta de regressão linear encontrada ( $P = -53.36 - 18.9d$ ) não representa minimamente os dados.

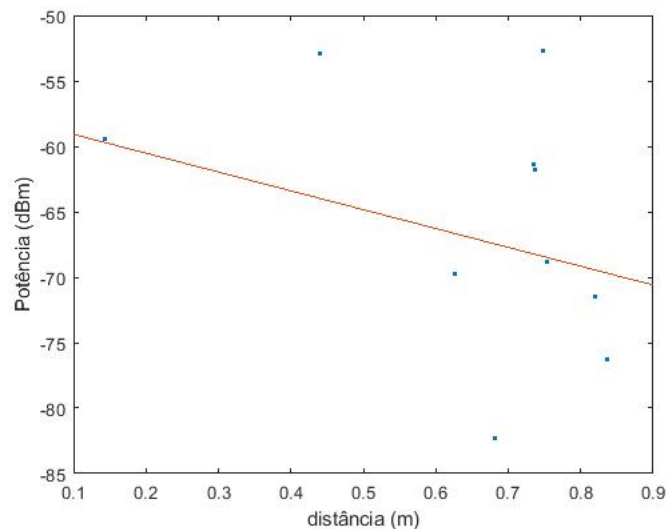
Métrica	Valor	Valor ideal
SSE	707.8	0
RMSE	9.4	0
R-Squared	0.17	1

**Tabela 4.1:** Métricas/indicadores e respectivos valores.

#### 4.2.3.B Regressão Linear (Pares)

Outra hipótese estudada foi a de a reta de regressão linear ser uma tal que conteria sempre dois pontos (do gráfico da Figura 4.9), à semelhança de um algoritmo RANdom SAMple Consensus (RANSAC)<sup>6</sup> [51]. Foram calculadas todas as combinações possíveis de pares de pontos e de seguida as mesmas métricas que na Secção 4.2.3.A, considerando os restantes pontos como os dados a adaptar. A melhor reta obtida, isto é, a que mostrou corresponder aos melhores indicadores, foi a reta  $P = -57.65 - 14.4d$  (ver Figura 4.10) e os indicadores são os presentes na Tabela 4.2.

<sup>6</sup>Após se ter obtido uma regressão linear que melhor aproxime os dados, dada a amostra de pontos utilizada, o algoritmo RANSAC pressupõe uma fase iterativa se existirem *outliers* - pontos cuja distância à reta de regressão seja maior do que um determinado *threshold*. No presente trabalho, esta fase não é tida em conta, uma vez que não se sabe que valores podem ser considerados *outliers* ou não.



**Figura 4.10:** "Reta de regressão linear".

Métrica	Valor	Valor ideal
SSE	718.3	0
RMSE	8.48	0
R-Squared	0.12	1

**Tabela 4.2:** Métricas/indicadores e respetivos valores.

Também a "reta de regressão" aqui encontrada não representa bem os dados. Uma explicação pode ser que, dada a natureza dos dados ser tão errática, o facto de dois pontos encaixarem no modelo (reta) encontrado não é nem remotamente relevante e é, sim, igualmente mau.

Posto isto, tornou-se evidente que seria necessário utilizar uma metodologia diferente para o tratamento das medições de potência.

## 4.3 Resultados

Nesta secção são apresentadas as metodologias utilizadas na solução final. Foram também estas alvo de experimentação e teste, mas provaram ser satisfatórias.

### 4.3.1 Limpeza, Completamento e Balizagem

#### 4.3.1.A Limpeza e Completamento

Como já referido, para se calcular a cada segundo a posição da *tag target*, é necessário que exista um valor de RSS e apenas um entre cada par de *tags*, correspondente a cada segundo do intervalo de

tempo em que se esteve a obter medições. Tal é verdadeiro quer seja para as *tags* âncoras ou para o *target*.

Para isso, com o algoritmo LCB (Algoritmo 4.1), definiram-se uma série de procedimentos a aplicar a cada vetor  $P_{ij}$  com as medições correspondentes a cada par de *tags*  $i$  e  $j$ . Todos os vetores foram ordenados em ordem ao tempo (ascendente).

Na fase de "limpeza" garantiu-se que só existia uma medição por instante de tempo  $t$  (segundo). No caso de existir mais do que uma<sup>7</sup> o valor de  $P_{ij}(t)$  a guardar foi a média entre elas.

Por outro lado, aconteceu que em certos instantes  $t$  não foi registada qualquer valor de potência. Então, na fase de "Completamento", esses valores foram preenchidos. Havia duas situações em que tal poderia ocorrer: no início, para  $t = 1$ , ou aleatoriamente em qualquer  $t$ . Caso ocorresse no início da contagem de tempo, os primeiros valores iguais a zero tomariam o primeiro valor diferente de zero. Se acontecesse o segundo caso, assumiu-se sempre que as potências teriam um comportamento "monotónico", isto é: dado  $P_{ij}(t \neq 0) = 0$ , se o valor diferente de zero imediatamente antes fosse menor que o valor diferente de zero imediatamente depois então, neste espaço de tempo, as potências aumentariam (monotonia crescente), caso contrário diminuiriam (monotonia decrescente). Posto isto, foi calculado o quociente entre a diferença entre estes dois últimos valores e a diferença de instantes entre eles. De seguida, aos valores iguais a zero adicionou-se o valor anterior e o referido quociente e

---

<sup>7</sup>Em todas as montagens, mesmo que não referidas nesta dissertação, não foram registadas mais do que duas medições por segundo.



continuou-se as iterações no vetor  $P_{ij}$ .

---

**Algoritmo 4.1: LCB**

---

```
Dados:  $P_{ij}$ , tempo, delta  
Resultado:  $P_{ij}$   
// Limpeza  
for cada instante de tempo  $t$  do  
| if existe mais do que um  $P_{ij}(t)$  then  
| |  $p = \text{média}(P_{ij}(t));$   
| end  
end  
// Completamento  
if  $P_{ij}(1) = 0$  then  
| igualar todos os  $P_{ij}(t)$  iguais a zero anteriores ao 1º  $P_{ij}(t)$  diferente de zero, ao 1º  $P_{ij}(t)$   
| diferente de zero;  
end  
 $c=0;$   
for cada instante de tempo  $t$  do  
| if  $P_{ij}(t) = 0$  then  
| | if  $P_{ij}(t-1) = 0$  then  
| | |  $c = c+1;$   
| | end  
| | else  
| | |  $c=2;$   
| | end  
| end  
| if  $P_{ij}(t) \neq 0$  e  $c \neq 0$  then  
| |  $diff = (P_{ij}(t) - P_{ij}(t-c))/c;$  while  $c$  do  
| | |  $P_{ij}(t-c-1) = P_{ij}(t-c-2) + diff$   
| | |  $c = c-1;$   
| | end  
| |  $c=0;$   
| end  
end  
// Balizagem  
for cada instante de tempo  $t$  do  
| if  $P_{ij}(t) - P_{ij}(t-1) > \text{delta}$  then  
| |  $P_{ij}(t) = \text{média}(P_{ij}(t), P_{ij}(t-1));$   
| end  
end
```

---

#### 4.3.1.B Balizagem

A *tag target* foi carregada por uma pessoa. Considerou-se, que em média, a velocidade de caminhada normal do ser humano não é superior a 1 m/s. No caso dos vetores de potências referentes a pares de *tags* 'âncora-target', por vezes, existiam variações grandes, maiores que 10 dB, de instante para instante, o que não parece correto. E porque não considerá-las *outliers*? Decidiu-se que para estes vetores em particular, não seriam eliminados os *outliers*. Dada a natureza dos dados ser tão

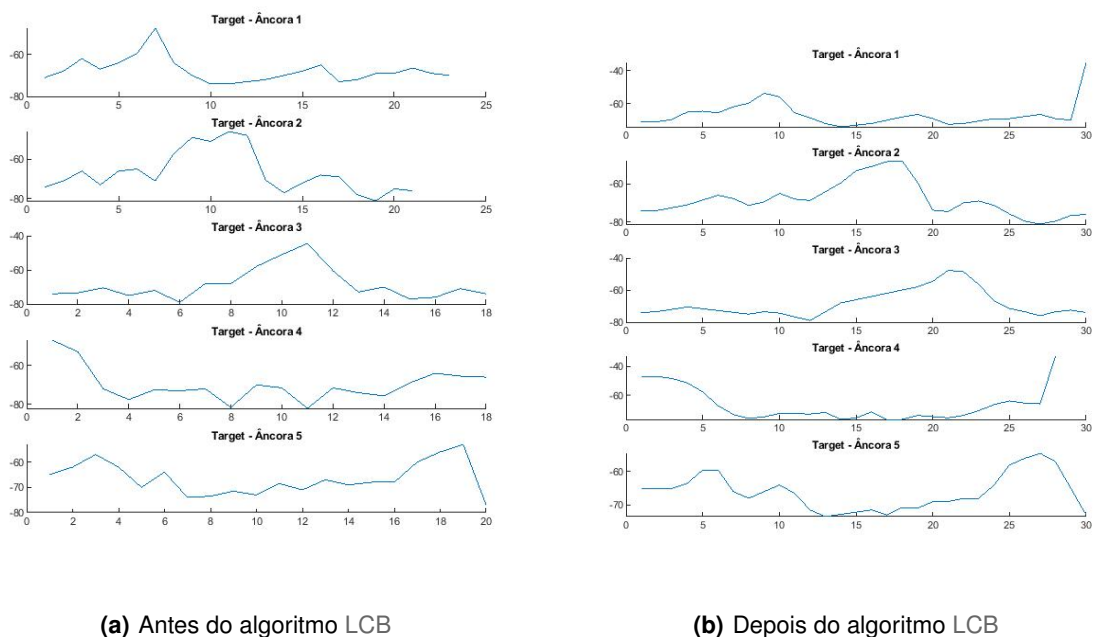
errática e poderem existir muitos dados "errados", considerou-se que embora estatisticamente um valor pudesse ser *outlier*, na realidade da montagem isso poderia não ser verdade; de certa forma, a eliminação de valores dessa forma poderia retirar "liberdade" aos valores verdadeiros de fazerem parte da solução, sejam eles quais forem.

Desta forma, encontrou-se a necessidade de ter em conta as condições reais da experiência e, sabendo a velocidade de caminhada, tentar de alguma forma limitar as variações. Para que isso fosse feito definiu-se uma variável *delta* que serviu de "baliza", ou seja, uma medição de RSS só poderia ser maior ou menor que a anterior *delta* unidades:

$$P_{ij}(t) \in [P_{ij}(t-1) - \text{delta}; P_{ij}(t-1) + \text{delta}] \quad (4.6)$$

Se o valor  $P_{ij}(t)$  não estivesse "dentro da baliza", dentro do intervalo de (4.6), tomaria a média entre o seu valor atual e o valor de  $P_{ij}(t-1)$ .

Após se experimentar vários valores, determinou-se que o valor de *delta* considerado ótimo foi de 2.5 dB e foi com este que se obtiveram os gráficos da Figura 4.11. Os gráficos em b) apresentam variações menos bruscas, estão mais "atenuados", em relação aos de a), mantendo uma forma que se assemelha à sua original, isto é, contêm variações (mudanças de monotonia) sensivelmente nos mesmos pontos.



**Figura 4.11:** Gráficos  $P(d)$  antes (a) e depois (b) da aplicação do algoritmo LCB (Algoritmo 4.1) - eixo horizontal: tempo(s); eixo vertical: potência(dBm)

### 4.3.2 Filtro de Kalman

Se por um lado não se quis eliminar completamente as variações nos vetores  $P_{ij}$  correspondentes a pares de *tags* 'âncora-target' mas apenas atenuar para que não fossem demasiado bruscas e "irreais", por outro lado entre os pares 'âncora-âncora' as variações podem ser inconvenientes. Uma vez que as âncoras estão imóveis seria de esperar que os níveis de RSS medidos se mantivessem dentro de um intervalo curto: como existe movimento nos espaços onde decorrem as atividades experimentais, a existência de valores de RSS constantes poderia perturbar os padrões de *multipath* no ambiente, mesmo entre sensores estáticos. Contudo, mesmo após a aplicação do Algoritmo 4.1, as medições obtidas por estes pares de sensores apresentam variações demasiado grandes (no que diz respeito a pares 'âncora-âncora'), pois estas estão contaminadas com ruído, fontes de atenuação e/ou amplificação e outras perdas não possíveis de quantificar. Posto isto, foi necessário realizar uma filtragem adicional aos valores de potência associados aos pares de âncoras, para que estes se aproximassem dos reais e fossem mais fiáveis que os obtidos pelo LCB. Um dos filtros mais utilizados para predição e aproximação de valores cujos valores reais são desconhecidos é o filtro de Kalman [52] [53].

Pode-se considerar o filtro utilizado como uma aproximação do de Kalman, na medida em que se assumiu um modelo dinâmico muito simples, com certas assumpções que podem ser apenas "aproximadamente verdadeiras", para representar um sistema estático.

Fundamentalmente, um filtro de Kalman divide-se em duas fases: predição e atualização. Pressupõe também um 'processo' (ou 'estado')  $x_k$  e uma 'medição' (ou 'observação')  $z_k$ , em que  $k$  indica o instante de tempo.

O estado  $x_k$  é descrito por

$$x_k = ax_{k-1} + bu_{k-1} + w_{k-1} \quad (4.7)$$

sendo  $a$  o modelo de transição de estados aplicado a  $x_{k-1}$  (relaciona o estado atual e o anterior),  $b_k$  a variável que relaciona o controlo de entrada (*control input*) com o estado  $x_{k-1}$  aplicada à entrada de controlo  $u_{k-1}$  e  $w_{k-1}$  é o ruído do processo e é assumido ter distribuição normal e covariância  $q$ ,  $w_k \sim \mathcal{N}(0, q)$ .

A cada instante de tempo  $k$  é obtida uma medição/observação  $z_k$

$$z_k = hx_k + v_k \quad (4.8)$$

tal que  $h$  é a variável que relaciona o estado real  $x_k$  com o observado  $z_k$  e  $v_k$  o ruído da observação que tem distribuição normal e covariância  $r$ ,  $v_k \sim \mathcal{N}(0, r)$ .

A fase de predição pode também ser chamada de 'estimativa *a priori*', uma vez que não depende do estado atual, como se vê em (4.9) e , em que  $\hat{\Sigma}_{apriori}$  é a covariância do erro.

$$\hat{x}_{apriori} = a\hat{x}_{k-1} + bu_{k-1} \quad (4.9)$$

$$\hat{\Sigma}_{apriori} = a\hat{\Sigma}_{k-1} + q \quad (4.10)$$

Na fase de atualização, ou fase de 'estimativa *a posteriori*', é necessário estimar o ganho ótimo de Kalman (que minimiza o erro da covariância *a posteriori*  $\hat{\Sigma}_k$ )  $K_k$  (4.11), bem como a estimação do estado (4.12),  $\hat{x}_k$ , e de  $\hat{\Sigma}_k$  (4.13).

$$K_k = \hat{\Sigma}_{apriori}h'(h\hat{\Sigma}_{apriori}h' + r)^{-1} \quad (4.11)$$

$$\hat{x}_k = \hat{x}_{apriori} + K_k(z_k - h\hat{x}_{apriori}) \quad (4.12)$$

$$\hat{\Sigma}_k = (1 - K_k h)\hat{\Sigma}_{apriori} \quad (4.13)$$

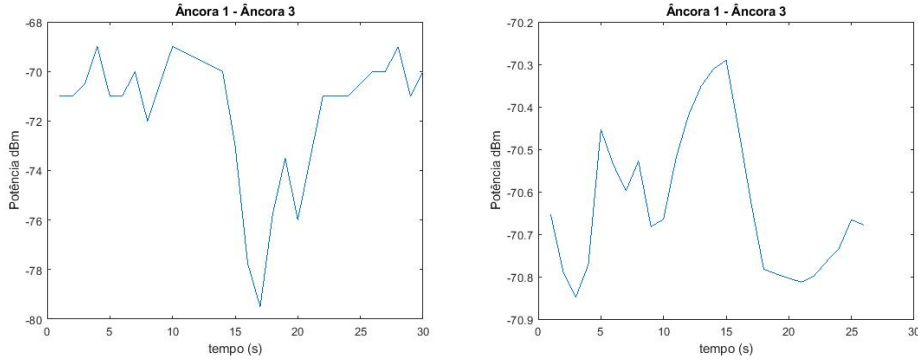
Considerando cada par de âncoras um sistema e que este não muda de instante para instante, assume-se que  $a = 1$  e que o único item trocado por cada elemento é a medição de RSS, não existindo controlo de entrada e  $u = 0$ . Tomou-se como variância do processo  $q = 10^{-5}$  e  $r$  a variância de cada vetor de potências entre âncoras. Para valores iniciais de estado e observação,  $\hat{x}_0$  e  $\hat{\Sigma}_0$ , tomaram-se a média das medições obtidas e 1, respetivamente.

Posto isto, a cada iteração, as equações da fase de predição simplificaram-se para as em (4.14) e as de atualização para as de (4.15).

$$\begin{cases} \hat{x}_{apriori} = \hat{x}_{k-1} \\ \hat{\Sigma}_{apriori} = \hat{\Sigma}_{k-1} + q \end{cases} \quad (4.14)$$

$$\begin{cases} k = \frac{\hat{\Sigma}_{apriori}}{\hat{\Sigma}_{apriori} + r} \\ \hat{x}_k = \hat{x}_{apriori} + k(z_{k-1} - \hat{x}_{apriori}) \\ \hat{\Sigma}_k = (1 - k)\hat{\Sigma}_{apriori} \end{cases} \quad (4.15)$$

A aplicação do filtro de Kalman, ainda que bastante simplificada, revelou-se proveitosa. Sendo que nesta montagem existem 10 pares de 'âncora-âncora' seria moroso apresentar todas comparações mas, a título de prova, na Figura 4.12 consegue-se comparar os valores de RSS medidos entre duas âncoras, a 1 e a 3, antes da aplicação do filtro (a) e depois (b). Apesar do gráfico obtido não ser uma reta horizontal constante, a variação que se observa é muito pequena ( $< 1dB$ ) pelo que os valores se podem considerar constantes, aproximando-se do que seria o ideal mas preservando ainda um certa componente experimental.



(a) Depois do LCB e antes do filtro de Kalman aproximado. (b) Depois do filtro de Kalman aproximado.

**Figura 4.12:** Medições entre as âncoras 1 e 3 antes e depois da aplicação do filtro de Kalman Aproximado.

### 4.3.3 Entrada do MDS

Uma vez "tratadas" e prontas todas as medições entre pares de sensores, pode-se partir para o algoritmo que utiliza MDS. Este algoritmo recebe de entrada uma matriz de dissimilaridades  $\Delta$  e cada entrada desta,  $\delta_{ij}$ , corresponde ao simétrico do valor de potência do par  $ij$ ,  $-P_{ij}$ , por exemplo, a dissimilaridade  $\delta_{13}$  será o simétrico do valor de potência medido entre as *tags* 1 e 3 ( $\Delta = -P_{ij}$ ).

$$P_{ij} = \begin{bmatrix} 0 & -60.71 & -70.65 & -68.65 & -61.57 & -71 \\ -60.71 & 0 & -53.03 & -68.26 & -76.17 & -74 \\ -70.65 & -53.03 & 0 & -81.80 & -52.72 & -74 \\ -68.65 & -68.26 & -81.80 & 0 & -59.48 & -47 \\ -61.57 & -76.17 & -52.72 & -59.48 & 0 & -65 \\ -71 & -74 & -74 & -47 & -65 & 0 \end{bmatrix} \quad (4.16)$$

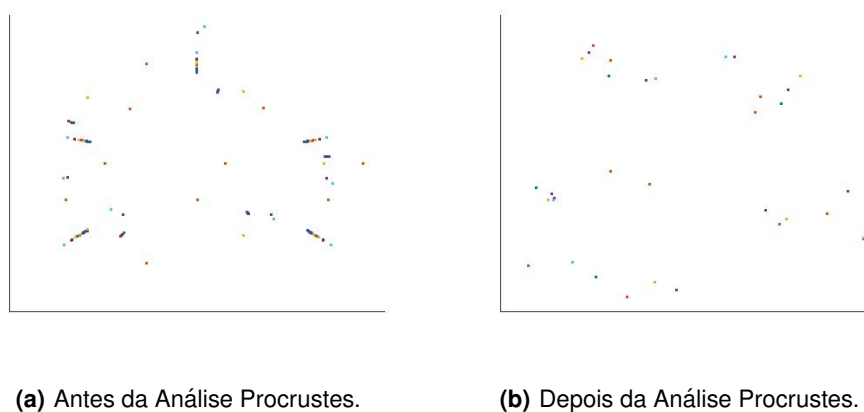
$$\Delta = \begin{bmatrix} 0 & 60.71 & 70.65 & 68.65 & 61.57 & 71 \\ 60.71 & 0 & 53.03 & 68.26 & 76.17 & 74 \\ 70.65 & 53.03 & 0 & 81.80 & 52.72 & 74 \\ 68.65 & 68.26 & 81.80 & 0 & 59.48 & 47 \\ 61.57 & 76.17 & 52.72 & 59.48 & 0 & 65 \\ 71 & 74 & 74 & 47 & 65 & 0 \end{bmatrix} \quad (4.17)$$

### 4.3.4 MDSCALE VS SMACoF

Para se obter, a cada instante de tempo, uma configuração de pontos,  $X$ , de onde seja possível retirar as coordenadas da *tag target* para se visualizar a sua trajetória, utilizaram-se os dois algoritmos descritos na Secção 3.1, SMACoF e MDSCALE. Ambos tiveram como entrada as mesmas matrizes de dissimilaridades  $\Delta$ .

#### 4.3.4.A Efeito da Análise Procrustes

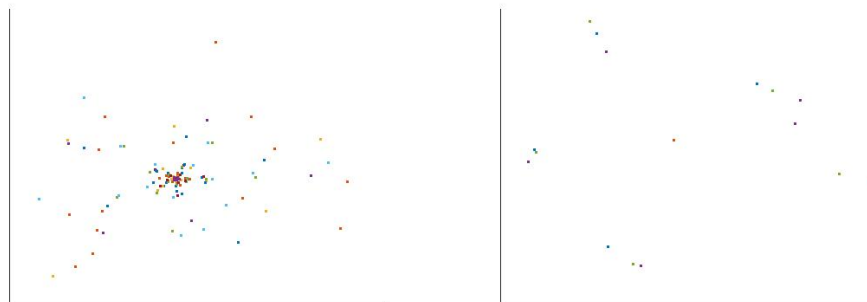
Primeiramente aplicaram-se os algoritmos a matrizes  $\Delta$  que apenas continham dissimilaridades entre âncoras. A razão deste passo na solução foi a necessidade de testar se a análise Procrustes (ilustrada na Secção 3.2) estaria efetivamente a contribuir para o melhoramento dos resultados. Analisando os "gráficos de pontos" da Figura 4.13, que corresponde ao algoritmo MDSCALE, observa-se que em 4.13(a), apesar de haver uma certa tendência para cinco zonas específicas (as 5 âncoras utilizadas), existem também vários pontos dispersos. Já na Figura 4.13(b) apesar de não haver uma densidade de pontos tão grande nas zonas que seriam das âncoras, não existe uma dispersão tão grande como no gráfico 4.13(a).



**Figura 4.13:** Representação das âncoras - algoritmo MDSCALE.

Já quando se utilizou o algoritmo SMACoF, cujos gráficos estão na Figura 4.14, a análise Procrustes fez uma diferença enorme: de uma amálgama de pontos, com grande densidade no centro da configuração, passou-se para alguns pontos dispersos em que, apesar de o resultado não ser tão bom como o do MDSCALE, se conseguem distinguir 5 zonas.

Após se verificar que, de facto, a análise Procrustes tinha algum efeito nas coordenadas (e posteriores configurações) e este sendo positivo, prosseguiu-se utilizando-a.



(a) Antes da Análise Procrustes.

(b) Depois da Análise Procrustes.

**Figura 4.14:** Representação das âncoras - algoritmo SMACoF.

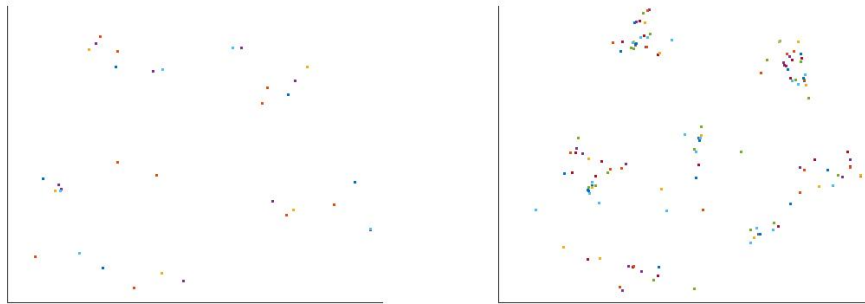
De notar que os dois gráficos que se encontram na Figura 4.13 estão na mesma escala, assim como os os da Figura 4.14. Porém, tal não é relevante uma vez que o que se pretende é ver os pontos onde o target se encontra e não conhecer as suas coordenadas. Caso a escala fosse diferente, ainda seria possível compará-los, dado que a diferença de escala resultaria na ampliação ou diminuição de um deles.

#### 4.3.4.B Impacto da Adição do Target

As *tags* que desempenham o papel de âncoras estão imóveis. Idealmente, essa característica transpareceria nas medições de RSS e por conseguinte nas configurações de pontos obtidas. O que já se verificou anteriormente é que tal não acontece: as coordenadas destes sensores também mudam de instante para instante. À parte disso, seria interessante e também importante que, com a adição das dissimilaridades dos pares 'target-âncora' as âncoras mantenham, ainda que não exatamente, a sua configuração. Só assim se poderá considerar minimamente fiável equacionar visualizar a trajetória de um target.

Nas Figuras 4.15 e 4.16 as coordenadas que são mostradas (a e b)) já sofreram análise Procrustes. Após observação destas, considerou-se que, mesmo não sendo iguais, continuam a dar a sensação de estarem pontos concentrados em 5 zonas distintas, numa disposição que faz lembrar as *tags*. Na verdade parece até existir uma maior demarcação daquelas áreas, o que pode indicar que um maior número de nós/*tags* melhora a qualidade da solução.

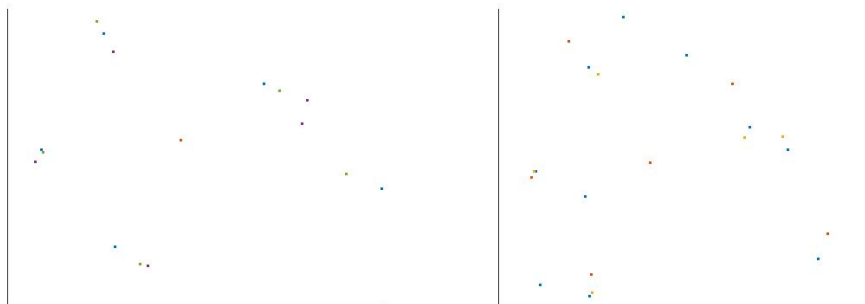
Decidiu-se, então prosseguir para o passo final.



(a) Antes da adição do target.

(b) Após a adição do target.

**Figura 4.15:** Âncoras - algoritmo MDSCALE.



(a) Antes da adição do target.

(b) Após a adição do target.

**Figura 4.16:** Âncoras - algoritmo SMACoF.

#### 4.3.4.C Análise Procrustes - Target

Tendo-se revelado uma ferramenta proveitosa para este trabalho, em particular, verificado o seu efeito de melhoramento nas coordenadas das âncoras, houve a intenção de estênde-la às coordenadas do *target* também. Isto constitui um problema já que a análise pressupõe uma configuração real para a qual outra vai ser adaptada e, no caso do *target*, não se sabe *a priori* a sua posição (situação contrária à das âncoras).

A solução que se encontrou para colmatar esta dificuldade foi, a cada instante de tempo, calcular a transformação que adaptava a configuração de âncoras obtida pelo MDS na sua configuração de âncoras real, e aplicá-la ao *target* (3.39). Obtiveram-se então os gráficos das Figuras 4.17 e 4.18. No da Figura 4.17 verifica-se que na área central não se encontram pontos, o que faz bastante sentido,



pois o *target* descreveu uma trajetória "à volta" das âncoras, nunca se encontrando na parte central do espaço. O gráfico da Figura 4.18 não parece tão consistente com a realidade, existe um ponto perto do centro. A razão deste gráfico ter poucos pontos prende-se de para algumas coordenadas este não ter conseguido encontrar um valor ótimo de *Stress*, o que levou a que estas fossem descartadas, tendo sobrado as visíveis no gráfico.



**Figura 4.17:** Posições do target obtidas pelo algoritmo MDSCALE.



**Figura 4.18:** Posições do target obtidas pelo algoritmo SMACoF.

#### 4.3.4.D Filtragem das distâncias

A "balizagem" aplicada aos valores de potência (ver Secção 4.3.1.B) tinha o objetivo de não retirar a liberdade aos dados, de não ser demasiado restritiva, uma vez que é complicado definir o que está certo ou errado quando se trata da métrica de RSS. Por isto mesmo, os valores obtidos através da filtragem com o filtro de Kalman também contêm erros.

Nesta fase, as coordenadas correspondentes aos pontos ocupados pelo *target* encontram-se já

calculadas. Em ordem a se obterem melhores resultados, executou-se uma filtragem sobre estas.

O filtro utilizado é quase outro tipo de balizagem, mas agora aplicado às coordenadas do *target*. Para este também se considerou a velocidade de caminhada do ser humano, de aproximadamente 1 m/s. Tal significa que entre instantes, a distância entre um ponto e o seu consecutivo não deveria ser maior que 1 m. No Algoritmo 4.2 é ilustrado que, se a condição anterior não acontecer, é feita uma média ponderada, em que o último ponto estimado  $\hat{X}_{target(t-1)}$  tem um peso quatro vezes superior ao ponto que se está a analisar  $X_{target(t)}$ . Antes disso assumiu-se que o ponto  $X_{target(1)}$  (primeiro ponto) estava correto, o que pode não ser verdade.

---

**Algoritmo 4.2:** Filtro das Distâncias

---

**Dados:**  $X_{target}(t)$ , tempo, velocidade

**Resultado:**  $\hat{X}_{target}(t)$

```

 $\hat{X}_{target}(t) = \text{vetor } 1 \times \text{tamanho}(\text{tempo}); \hat{X}_{target}(1) = X_{target}(1); \text{ for cada instante } t \text{ do}$ 
  | if  $|\hat{X}_{target}(t-1) - X_{target}(t)| > \text{velocidade}$  then
  |   |  $\hat{X}_{target}(t) = \frac{(4 * \hat{X}_{target}(t-1) + X_{target}(t))}{5};$ 
  | end
  | else
  |   |  $\hat{X}_{target}(t) = X_{target}(t);$ 
  | end
end

```

---

Posto isto, obtiveram-se as trajetórias que se podem ver na Figura 4.19. Ainda que não perfeitamente, a trajetória da Figura 4.19(a), cujo algoritmo de MDS foi o MDSCALE, descreve uma forma muito parecida a um quadrado: começa com algo que parecem ser inversões de marcha<sup>8</sup> mas depois segue para algo que se aproxima do caminho real. Já a trajetória da Figura 4.19(b) (correspondente ao algoritmo SMACoF) ficou um pouco aquém, parece uma deslocação na diagonal, o que não aconteceu na realidade. A razão para estes resultados pode estar no que já foi observado para a Figura 4.18: o algoritmo SMACoF não encontrou um valor ótimo de *Stress* em todas as suas iterações, sendo as coordenadas do *target* correspondentes descartadas; tal resultou em poucos pontos, o que é mais um indicador de que o SMACoF não teve uma performance tão boa quanto o MDSCALE.

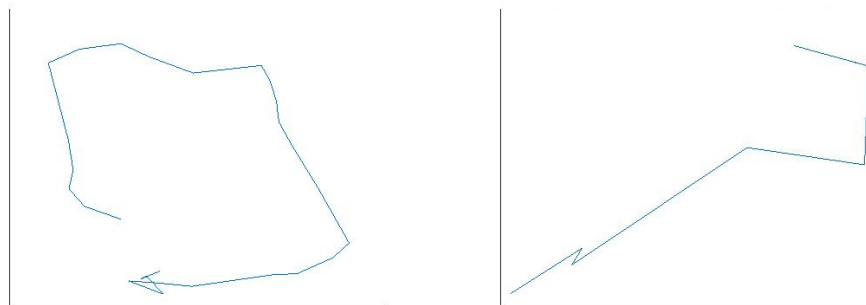
Contudo, é de salientar que só houveram duas "inversões de marcha" e estas parecem ter sido pequenas, pelo que poderá ser, pelo menos, um indicador de direção.

### 4.3.5 MDSCALE - Efeitos do ambiente

Como já referido no decorrer desta dissertação, nos ambientes físicos reais (não ideais), existem muitos fatores atenuantes no que diz respeito à propagação do sinal. Tal acontece quer o emissor e o recetor estejam em LoS ou não.

---

<sup>8</sup>A montagem em questão foi realizada por uma pessoa só, e para obtenção das medições é necessário iniciar o firmware. Fisicamente implicou a pega no sensor e rotação, um movimento rápido.



(a) Trajetória - algoritmo MDSCALE.

(b) Trajetória - algoritmo SMACoF.

**Figura 4.19:** Trajetórias do Target.

Foram realizadas várias montagens e em ambientes/espços físicos diferentes. Estas produziram resultados diferentes quer o número de âncoras utilizado fosse igual, superior ou inferior.

De facto, dentro do mesmo ambiente nunca foram alcançados resultados iguais pois, apesar do espaço físico poder ser o mesmo (mesmas paredes, mesma disposição de mobília/objectos, etc.), existem elementos no meio que podem variar. Entre estes últimos estão pessoas que se podem mover livremente no espaço e correntes de ar, não se descartando a hipótese de existirem muitos mais. De notar que em todas as montagens nas quais figura um *target* existe sempre uma pessoa no meio (pois esta *tag* é carregada por ela) e sendo o corpo humano constituído por 60% de água [54] constitui portanto uma fonte de atenuação.

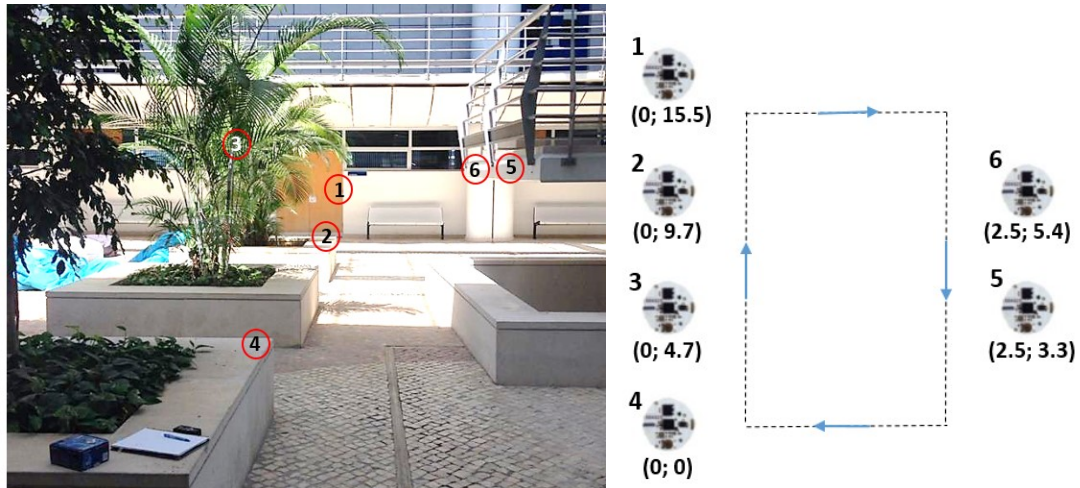
Como anunciado na Secção 4.2.1, a orientação das *tags* é um fator de máxima importância, uma vez que é conhecido que a direção de ganho máximo é a ortogonal ao plano da antena. Nas montagens, raramente se conseguiu colocar/implementar todos os sensores à mesma altura. Tal não é necessariamente mau, em ambientes reais esta implementação também poderá ser complicada devido à própria infraestrutura do espaço e/ou necessidades que as *tags* utilizadas requeiram (como ligação a uma fonte de eletricidade ou exposição solar, etc.). Contudo, as *tags* foram sempre colocadas com a antena na vertical, o que não deve constituir problemas em qualquer ambiente.

#### 4.3.5.A Átrio do Campus TagusPark

No átrio do *campus* do TagusPark foi realizada a montagem "h04", na qual foram dispostas seis âncoras tal como se mostra na Figura 4.20(a). Foi considerado um "espaço virtual"<sup>9</sup> de 2.5 m por 15.5 m. A âncora 3 encontra-se implementada numa árvore, a 2 m em relação ao solo. Como vários trabalhos indicam [55] [56] [57], as árvores, mais propriamente a sua folhagem, são um forte atenuador

<sup>9</sup>Entenda-se por espaço virtual o perímetro da configuração das âncoras.

à propagação de sinais. Porém, na prática permitem "esconder" os sensores dos olhos dos utilizadores, o que poderá ser um requisito em certos ambientes. As âncoras 2 e 4 estão a uma altura de 50 cm e as *tags* 1, 5 e 6 a 1.6 m do chão. O *target* moveu-se a cerca de 1.45 m do chão, descrevendo uma trajetória como indicado na Figura 4.20(b).



(a) Disposição das âncoras - Átrio do TagusPark.

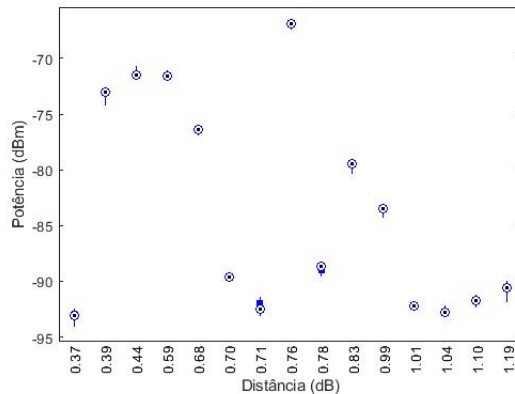
(b) Trajetória descrita pelo *target* entre as âncoras (de 1 a 6) e coordenadas destas.

**Figura 4.20:** Montagem no átrio do TagusPark.

Tal como se tinha observado na Figura 3.1, também o gráfico na Figura 4.21 não permite identificar uma relação clara entre as medições de potência entre cada par de âncoras e a distância entre elas. No ambiente desta experiência pode-se considerar que ocorreu um deterioramento na qualidade dos dados adquiridos, visto que estes se apresentam mais erráticos. Tal dever-se-á à presença mais acentuada de fenómenos de *multipath* e *fading*, causados quer pelos elementos imóveis no meio quer pelos móveis (pessoas e pombos a deslocarem-se entre as âncoras).

Apesar de ainda estar afastada da realidade, a introdução de mais uma *tag* nos cálculos (neste caso, o *target*) permitiu a visualização de uma constelação de 6 âncoras (Figura 4.22(b)). Considerando apenas as medições entre as âncoras, obteve-se a da Figura 4.22(a), na qual parecem constar somente 4 âncoras. Dada a grande quantidade de perturbações encontradas neste ambiente, apesar de as âncoras se encontrarem ao alcance umas das outras isso pode não ser suficiente para encontrar a sua localização; uma vez que o *target* passa mais perto delas, efetua medições que podem ser consideradas mais fiáveis (como está a uma menor distância existem menos perdas entre o emissor e o recetor), o que leva a melhores resultados.

Dada a trajetória real (Figura 4.20(b)), a obtida através do algoritmo MDSCALE e após filtragem das distâncias, Figura 4.23, não se assemelha muito àquela. Porém, o sentido do deslocamento parece ter



**Figura 4.21:** Distância *versus* Potência - montagem h04.



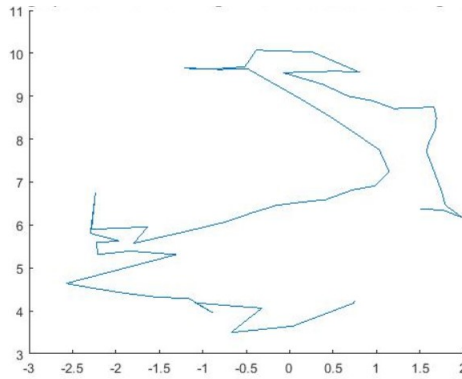
**(a)** Coordenadas das âncoras - antes da introdução do *target*. **(b)** Coordenadas das âncoras - depois da introdução do *target*.

**Figura 4.22:** Coordenadas das âncoras - montagem no átrio do TagusPark.

seja frequentemente respeitado (de acordo com as setas a azul da Figura 4.20(b)), ainda que o mesmo não tenha acontecido com a mesma frequência com a direção, comparativamente.

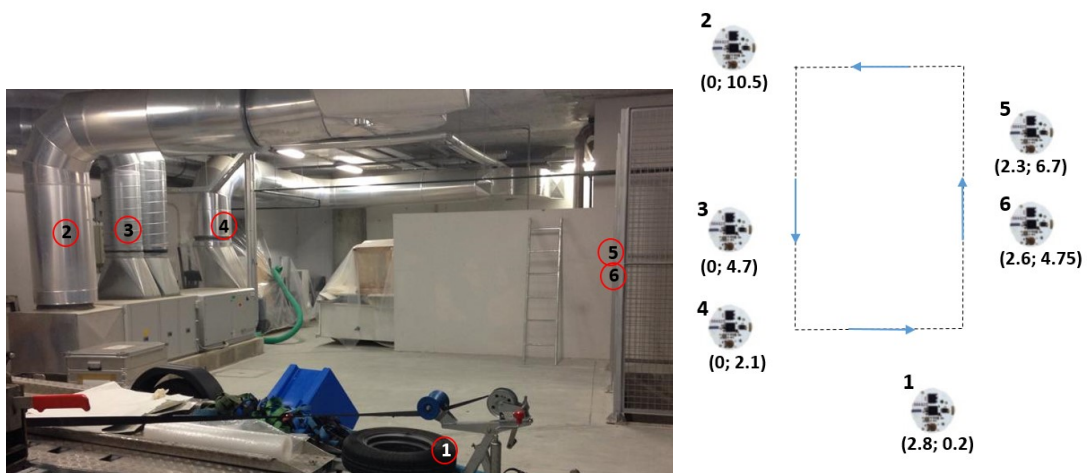
#### 4.3.5.B Garagem do Campus TagusPark

Na garagem do *campus* do TagusPark foi conduzida, entre outras, a montagem "k02". Neste sítio não existia conexão à rede wi-fi nem dados móveis e à altura da realização da montagem, estavam lá várias estruturas metálicas, muitas das quais bem polidas, como se observa na Figura 4.24(a). Apesar de no início se ter pensado que a condução de uma experiência neste ambiente permitiria verificar se a propagação dos sinais emitidos pelos sensores sofrem (de forma significativa) interferências de outros sinais, dada a quantidade de estruturas metálicas e reflexões de sinais que daí podem advir não torna claro um veredito. Contudo, na montagem "j01" (utilizada nas subsecções anteriores) também estavam



**Figura 4.23:** Trajetória descrita pelo *target* (MDSSCALE) - átrio do TagusPark.

presentes aquele tipo de estruturas, ainda que em menor número, pelo que se decidiu prosseguir com a análise dos resultados da presente montagem.



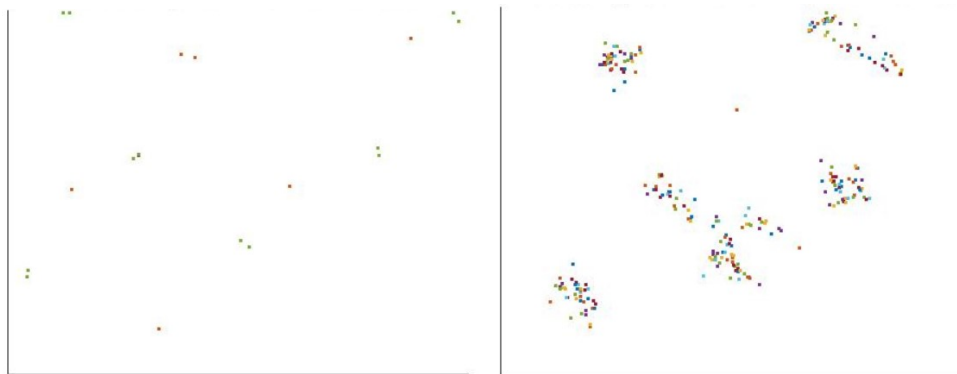
**(a)** Disposição das âncoras - Garagem do TagusPark.

**(b)** Trajetória descrita pelo *target* entre as âncoras (de 1 a 6) e coordenadas destas.

**Figura 4.24:** Montagem na garagem do TagusPark.

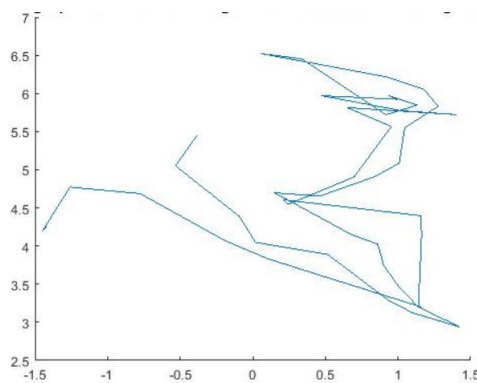
Ao contrário do que aconteceu na montagem no átrio, nesta as âncoras encontram-se melhor identificadas, ainda que a estimativa de constelação formada por elas, representada na Figura 4.25(b), diste da realidade.

Surpreendentemente, a trajetória obtida está longe da que foi realmente descrita. Seria preciso uma "grande dose de boa vontade" para encontrar, na Figura 4.26 semelhanças em termos de sentido e direção no que toca à trajetória real (figura 4.24(b)). Apesar de não existir conexão à rede wi-fi na garagem, a propagação dos sinais RF foi sem qualquer dúvida fortemente afetada por outros fenómenos e/ou interferências.



(a) Coordenadas das âncoras - antes da introdução do *target*. (b) Coordenadas das âncoras - depois da introdução do *target*.

**Figura 4.25:** Coordenadas das âncoras - montagem na garagem do TagusPark.



**Figura 4.26:** Trajetória descrita pelo *target* (MDSCALE) - garagem do TagusPark.

Verificou-se que o ambiente na garagem não é propício à localização de um móvel, utilizando os métodos expostos nesta dissertação. Por outro lado, a localização de sensores imóveis poderia ser minimamente viável, uma vez que a determinação da localização das âncoras não ficou tão aquém do aceitável. Na prática, este ambiente é parecido ao existente nalguns meios industriais (com muitas estruturas metálicas), nos quais algumas vezes é necessário conhecer a localização de determinados objectos (estando estes imóveis, em prateleiras, por exemplo).

# 5

## Conclusão

### Conteúdo

---

5.1	Apreciação Global . . . . .	61
-----	-----------------------------	----

---





## 5.1 Apreciação Global

Para chegar a este ponto do trabalho, pouco se teve em conta o Stress na avaliação dos resultados que se foram obtendo. Tal aconteceu porque, apesar do mesmo ser um indicador de "goodness-of-fit" não é um indicador absoluto. Aliás, na literatura é várias salientado isso mesmo: um stress baixo, menor que 1% [58], não indica necessariamente uma boa solução MDS [59] ou que aquela seja a melhor de todas as iterações feitas. Há sempre que ter em conta os dados que estão a ser estudados, pois se se sabe de antemão que estes contêm muitos erros, como acontece neste trabalho, não será boa ideia guiarmo-nos exclusivamente por uma medida que depende deles próprios.

Verificou-se uma melhor performance utilizando o algoritmo MDSCALE, no que diz respeito à qualidade dos resultados. Tendo em atenção a montagem "j01", a trajetória obtida (representada na Figura 4.19(a)) assemelha-se bastante à real, o que é bastante bom tendo em conta a enorme quantidade de ruído contida nas medições de RSS entre cada par de sensores (sejam âncoras ou targets).

No que diz respeito ao algoritmo SMACoF, à semelhança do MDSCALE, a cada segundo seria suposto ele convergir para a solução (configuração de pontos) cujo valor de *Stress* seria mínimo. O que pode ter acontecido, devido à fraca qualidade dos dados (medições de RSS), é que este mínimo seria local e não global, levando a soluções não ótimas. Também aconteceu o SMACoF não ter conseguido convergir para um mínimo, sendo as coordenadas correspondentes descartadas, o que resultou na pouca quantidade de pontos e, por conseguinte, na fraca qualidade do resultado final.

Outra apreciação a ter em conta é a de como ambientes diferentes permitem obter qualidades de resultados diferentes. Na Tabela 5.1 encontram-se resumidas as características dos espaços onde foram realizadas as montagens referidas nesta dissertação. A montagem com que se obteve melhores resultados foi a "j01", o que pode indicar que a "sala do tanque" serve como espaço referência para o sistema utilizado. Já com a montagem "k02" obtiveram-se os piores resultados, o que pode ser sinal de que um espaço com uma presença tão forte de estruturas metálicas como a garagem do TagusPark, não é o indicado para os propósitos pretendidos nesta dissertação.

**Tabela 5.1:** Ambientes onde foram realizadas as montagens descritas nesta dissertação.

Montagem	Local	Condições
j01	Laboratório do ISR (sala do tanque)	Presença de estruturas metálicas (polidas e não polidas), vidros e pessoa que carregava o <i>target</i> ; espaço amplo/"desafogado"
h04	Átrio do TagusPark	Presença de árvores, estruturas em cimento, pessoas (para além da que carregava o <i>target</i> ) e pombo
k02	Garagem do TagusPark	Presença acentuada de estruturas metálicas (polidas e não polidas) e pessoa que carregava o <i>target</i> ; espaço muito preenchido (pelas estruturas)/pouco "desafogado"

### 5.1.1 Conclusão

Um aspecto bastante interessante foi verificar que o aumento do número de sensores (adição do target) ajudou na procura de uma melhor configuração. Tal corroborou com alguma literatura encontrada, onde se afirma que o Stress diminui e é mais fácil de minimizar com o aumento do número de nós.

A cada passo mais próximo da solução final foi-se encontrando motivos para justificar o uso de certas metodologias (como o teste à análise Procrustes, na Secção 4.3.4.A). Motivos e não razões porque, como se verificou ao longo desta dissertação, no início através da literatura e depois através do trabalho experimental, um sistema que assente na métrica RSS é por si só um sistema complicado. A única explicação encontrada para a relação (ou a inexistência dela) das medições obtidas com as distâncias, é que estes sinais de RF são fortemente afetados pelo meio, ficando os dados de RSS bastante erráticos, o que torna difícil a criação de filtros adequados. Por outro lado, com o *boom* de dispositivos que atualmente ocupam a banda ISM, pode haver também aqui mais um aumento de interferências para além das mais óbvias, presentes no meio (obstáculos, ocasional NLoS, fenómenos de multipath e fading, entre outros). Por essa razão, na realização das montagens tentou-se sempre colocar todos os sensores em LoS.

Como seria de esperar, obtiveram-se melhores resultados num ambiente mais calmo e "desafogado", com uma menor quantidade de mobílias e outras estruturas e pessoas.

Em comparação com a experiência "Vírus", que deu o mote a esta dissertação e se encontra esquematizada na Figura 4.3, o trabalho desenvolvido pela Thales poderia estar mais virado para problemas de proximidade, enquanto que o âmbito do presente trabalho se prende mais com localização. Assim sendo, o balanço foi positivo.

### 5.1.2 Trabalho futuro

Existem várias hipóteses a ter em conta como trabalho futuro. A mais evidente é realizar a parte experimental com outro equipamento, nomeadamente outros sensores. Provavelmente com outros sensores mais potentes e que suportem Wi-Fi, as medições poderiam ser melhores (se sofrerem menos interferências).

Existem inúmeras metodologias disponíveis, não só na área da engenharia e matemática, mas também noutras áreas, que se podem adaptar e aplicar ao problema em estudo. É importante pensar "fora da caixa". Assim, poder-se-á, com o mesmo equipamento, testar novos filtros, inventados ou daqueles já disponíveis na literatura, que permitam aproximar as medições de RSS ao que é de esperar que sejam os valores reais assim, obter outras soluções que melhor se adaptem à realidade.

Seria interessante acrescentar mais âncoras e mais targets às montagens, em ordem a verificar se

estes "cooperariam" na diminuição de Stress e melhores configuração ou se a qualidade se manteria.

Do ponto de vista teórico, conhecendo o diagrama de propagação das antenas dos sensores (4.2.1), poder-se-á explorar os dados de forma a verificar (ou não) a existência de uma relação entre as distâncias "âncora-chão" e "target-chão". Por outras palavras, uma vez que o diagrama de propagação de cada antena corresponde ao de um dipolo magnético, poder-se-á conseguir relacionar o ganho da antena e/ou ângulo de transmissão com a distância entre duas antenas, partindo do princípio que a altura das duas tags ao chão é conhecida.

Por último, o aspeto mais importante será o delineado da trajetória (localização) em tempo real. De facto, nenhuma das metodologias utilizadas tinha qualquer entrave a uma abordagem em tempo real, mas a avaliar pela qualidade dos resultados não seria uma boa ideia. Contudo, no futuro, certamente irá surgir um sistema de localização indoor para as massas.

# Bibliografia

- [1] (2020, Junho) Thales. [Online]. Available: <https://www.thalesgroup.com/en>
- [2] (2020, Julho) Open Beacon. [Online]. Available: <https://www.openbeacon.org/device.html>
- [3] V. Rato, "RSS-based Indoor Localization using a Network of Open-Source Wireless Tags," Master's thesis, Instituto Superior Técnico, May 2018.
- [4] A. Malik, *RTLS for Dummies*. John Wiley & Sons, 2009.
- [5] C. Hoene and J. Willmann, "Four-way TOA and software-based trilateration of IEEE 802.11 devices," in *2008 IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications*. IEEE, 2008, pp. 1–6.
- [6] I. Borg, P. J. Groenen, and P. Mair, *Applied multidimensional scaling*. Springer Science & Business Media, 2012.
- [7] (2018, Junho) Wifarer - Indoor Positioning. [Online]. Available: <http://www.wifarer.com/museums.html>
- [8] P. Misra, B. P. Burke, and M. M. Pratt, "GPS performance in navigation," *Proceedings of the IEEE*, vol. 87, no. 1, pp. 65–85, 1999.
- [9] (2018, Junho) Skyhook Wireless. [Online]. Available: <https://www.skyhookwireless.com/>
- [10] A. Mesmoudi, M. Feham, and N. Labraoui, "Wireless sensor networks localization algorithms: A comprehensive survey," *arXiv preprint arXiv:1312.4082*, 2013.
- [11] J. A. Costa, N. Patwari, and A. O. Hero III, "Distributed weighted-multidimensional scaling for node localization in sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 2, no. 1, pp. 39–64, 2006.
- [12] W. Shi and V. W. Wong, "MDS-based localization algorithm for RFID systems," in *2011 IEEE International Conference on Communications (ICC)*. IEEE, 2011, pp. 1–6.

- [13] M. A. Koledoye, T. Facchinetti, and L. Almeida, "Improved MDS-based localization with non-line-of-sight RF links," *Journal of Intelligent & Robotic Systems*, vol. 98, no. 1, pp. 227–237, 2020.
- [14] S. Paulose, E. Sebastian, and B. Paul, "Acoustic source localization," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 2, no. 2, pp. 933–9, 2013.
- [15] (2020, Junho) Active Bat. [Online]. Available: <http://www.cl.cam.ac.uk/research/dtg/attarchive/bat/>
- [16] (2020, Junho) Cricket. [Online]. Available: <http://cricket.csail.mit.edu>
- [17] V. F. S. F. M. Santos and L. Almeida, "Auto-localização em pequenos robôs móveis e autônomos: O caso do robô Bulldozer IV," *Electrónica e Telecomunicações*, vol. 3, no. 6, pp. 548–556, 2002.
- [18] B. Ferreira, "Hybrid Indoor Localization Based on Ranges and Video," Master's thesis, Instituto Superior Técnico, October 2014.
- [19] R. D. Cruz and J. R. Pedrasa, "RSSI-based localization in an emulated active RFID system," in *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. IEEE, 2019, pp. 0771–0778.
- [20] A. F. Molisch, "Ultrawideband propagation channels-Theory, measurement, and modeling," *IEEE transactions on vehicular technology*, vol. 54, no. 5, pp. 1528–1545, 2005.
- [21] B. P. Crow, I. Widjaja, J. G. Kim, and P. T. Sakai, "IEEE 802.11 wireless local area networks," *IEEE Communications magazine*, vol. 35, no. 9, pp. 116–126, 1997.
- [22] A. Maratea, G. Salvi, and S. Gaglione, "Bagging to improve the calibration of RSSI signals in bluetooth low energy (BLE) indoor distance estimation," in *2019 15th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*. IEEE, 2019, pp. 657–662.
- [23] D. Niculescu and B. Nath, "Ad hoc positioning system (aps) using AOA," in *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428)*, vol. 3. IEEE, 2003, pp. 1734–1743.
- [24] P. Pettinato, N. Wirström, J. Eriksson, and T. Voigt, "Multi-channel two-way time of flight sensor network ranging," in *European Conference on Wireless Sensor Networks*. Springer, 2012, pp. 163–178.
- [25] K. Mahender, T. A. Kumar, and K. Ramesh, "Analysis of multipath channel fading techniques in wireless communication systems," in *AIP Conference Proceedings*, vol. 1952, no. 1. AIP Publishing LLC, 2018, p. 020050.

- [26] P. Liu, J. Yan, H. Hao, and Y. K. Wu, "Phase retrieval for short wavelength orbital angular momentum beams using knife-edge diffraction," *Optics Communications*, p. 126077, 2020.
- [27] D. Cahyono, F. Sheikh *et al.*, "Deterministic approach of indoor room THz multipath channel model," in *2020 8th International Conference on Information and Communication Technology (ICoICT)*. IEEE, 2020, pp. 1–5.
- [28] H. Will, T. Hillebrandt, and M. Kyas, "The FU berlin parallel lateration-algorithm simulation and visualization engine," in *2012 9th Workshop on Positioning, Navigation and Communication*. IEEE, 2012, pp. 131–136.
- [29] E. Niewiadomska-Szynkiewicz, "Localization in wireless sensor networks: Classification and evaluation of techniques," *International Journal of Applied Mathematics and Computer Science*, vol. 22, no. 2, pp. 281–297, 2012.
- [30] U. Flick, "Triangulation," in *Empirische Forschung und Soziale Arbeit*. Springer, 2011, pp. 323–328.
- [31] S. M. Kay, *Fundamentals of statistical signal processing, volume I: Estimation theory (v. 1)*. Prentice Hall PTR, 1993.
- [32] H. Lu, S. Mazuelas, and M. Z. Win, "Ranging likelihood for wideband wireless localization," in *2013 IEEE International Conference on Communications (ICC)*. IEEE, 2013, pp. 5804–5808.
- [33] A. Beck, P. Stoica, and J. Li, "Exact and approximate solutions of source localization problems," *IEEE Transactions on signal processing*, vol. 56, no. 5, pp. 1770–1778, 2008.
- [34] Y. Yao, P. Frasconi, and M. Pontil, "Fingerprint classification with combinations of support vector machines," in *International conference on audio-and video-based biometric person authentication*. Springer, 2001, pp. 253–258.
- [35] K. Wen, C. K. Seow, and S. Y. Tan, "An indoor localization and tracking system using successive weighted RSS projection," *IEEE Antennas and Wireless Propagation Letters*, 2020.
- [36] F. Chan, Y.-T. Chan, and R. Inkol, "Path loss exponent estimation and rssi localization using the linearizing variable constraint," in *MILCOM 2016-2016 IEEE Military Communications Conference*. IEEE, 2016, pp. 225–229.
- [37] C. Li, X. Huang, and L. Tang, "Rssi localization algorithm based on virtual sample nodes in wsn," in *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. IEEE, 2017, pp. 615–618.
- [38] S. Tomic, M. Beko, and R. Dinis, "Distributed RSS-AoA based localization with unknown transmit powers," *IEEE Wireless Communications Letters*, vol. 5, no. 4, pp. 392–395, 2016.

- [39] I. Borg and P. J. Groenen, *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media, 2005.
- [40] D. R. Hunter and K. Lange, “A tutorial on MM algorithms,” *The American Statistician*, vol. 58, no. 1, pp. 30–37, 2004.
- [41] F. Busing, J. J. Commandeur, W. J. Heiser, W. Bandilla, F. Faulbaum *et al.*, “Proxscal: A multidimensional scaling program for individual differences scaling with constraints,” *Softstat*, vol. 97, pp. 67–74, 1997.
- [42] C.-y. Miao, G.-y. Dai, K.-j. Mao, Y.-d. Li, and Q.-z. Chen, “Ri-MDS: Multidimensional scaling iterative localization algorithm using RSSI in wireless sensor networks,” *International Journal of Distributed Sensor Networks*, vol. 11, no. 11, 2015.
- [43] (2020, Julho) Mdscale. [Online]. Available: <https://www.mathworks.com/help/stats/mdscale.html>
- [44] A. Singh, “Conjugate gradient descent,” Carnegie Mellon, School of Computer Science, 2020.
- [45] J. C. Gower, “Generalized procrustes analysis,” *Psychometrika*, vol. 40, no. 1, pp. 33–51, 1975.
- [46] (2020, Julho) nrf51822. [Online]. Available: <https://www.nordicsemi.com/Products/Low-power-short-range-wireless/nRF51822>
- [47] M. Meriac, “Low power to the people,” OpenBeacon.
- [48] (2020, Julho) meriac/openbeacon-ng. [Online]. Available: <https://github.com/meriac/openbeacon-ng>
- [49] (2020, Julho) Laboratório DSOR. [Online]. Available: [https://welcome.isr.tecnico.ulisboa.pt/projects\\_cat/dsor/](https://welcome.isr.tecnico.ulisboa.pt/projects_cat/dsor/)
- [50] (2020, Julho) Evaluating goodness of fit. [Online]. Available: <https://www.mathworks.com/help/curvefit/evaluating-goodness-of-fit.html>
- [51] K. G. Derpanis, “Overview of the RANSAC algorithm,” *Image Rochester NY*, vol. 4, no. 1, pp. 2–3, 2010.
- [52] G. Welch, G. Bishop *et al.*, “An introduction to the Kalman filter,” Department of Computer Science, University of North Carolina, 1995.
- [53] A. Barrau and S. Bonnabel, “The invariant extended Kalman filter as a stable observer,” *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 1797–1812, 2016.



- [54] (2020, Julho) Água no corpo humano. [Online]. Available: <https://www.msmanuals.com/pt/casa/dist%C3%BArbios-hormonais-e-metab%C3%B3licos/equil%C3%ADbrio-h%C3%ADrico/a-%C3%A1gua-no-corpo>
- [55] M. Khatun, C. Guo, D. Matolak, and H. Mehrpouyan, "Indoor and outdoor penetration loss measurements at 73 and 81 GHz," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–5.
- [56] G. Durgin, T. S. Rappaport, and H. Xu, "Measurements and models for radio path loss and penetration loss in and around homes and trees at 5.85 GHz," *IEEE Transactions on Communications*, vol. 46, no. 11, pp. 1484–1496, 1998.
- [57] U. M. Qureshi, F. K. Shaikh, Z. Aziz, S. M. Z. S. Shah, A. A. Sheikh, E. Felemban, and S. B. Qaisar, "RF path and absorption loss estimation for underwater wireless sensor networks in different water environments," *Sensors*, vol. 16, no. 6, p. 890, 2016.
- [58] J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, vol. 29, no. 1, pp. 1–27, 1964.
- [59] W. Wagenaar and P. Padmos, "Quantitative interpretation of stress in Kruskal's multidimensional scaling technique," *British Journal of Mathematical and Statistical Psychology*, vol. 24, no. 1, pp. 101–110, 1971.

