

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



Comunicação e Coordenação em Rede de Veículos Aquáticos

André Filipe de Castro Reis

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador: Professor Aníbal Matos

27 de Outubro de 2013

Resumo

À medida que a tecnologia vai avançando, o Homem fica cada vez mais dependente que as máquinas consigam desempenhar tarefas de forma autónoma. Esta necessidade implica, no caso de várias máquinas envolvidas, que estas tenham sempre a capacidade de comunicação umas com as outras.

Este trabalho está inserido no âmbito do grupo da Faculdade de Engenharia da Universidade do Porto, OceanSys, que tem, como objectivo o desenvolvimento de sistemas avançadas com a capacidade de, automaticamente, recolher e processar dados em ambientes aquáticos, e pretende desenvolver um meio de assegurar que estas comunicações se realizam, através de algoritmos e simulações.

Abstract

As technology advances, Man develops a need of machinery that can perform tasks in an autonomous way. This need implies, in the case of multiple machines involved, that they have the ability to communicate with one another.

This work is inserted in the context of the working group from Faculdade de Engenharia da Universidade do PortoOceanSys, which has the purpose of developing advanced systems with the ability to, automatically, collect and process data in aquatic environments, and pretends to assure that these communications take place, through the use of algorithms and simulations.

Agradecimentos

Agradeço, primeiramente, aos meus pais pelo apoio dado durante todos os anos que estudei.

Agradeço, também à Joana pela paciência demonstrada quando as coisas corriam menos bem e precisava de alguém para desabafar.

Um agradecimento especial ao Professor Aníbal Matos pela orientação prestada no decorrer deste trabalho e, também, aos elementos pertencentes ao OceanSys pela disponibilidade demonstrada para ajudar a compreender o funcionamento dos ASV.

Por fim, agradeço a Matt Fig e a Paulo Silva pelo código que disponibilizaram *online* e que auxiliou à execução do trabalho.

André Reis

*“An example I often use to illustrate the reality of vanity, is this:
look at the peacock; it’s beautiful if you look at it from the front.
But if you look at it from behind, you discover the truth...
Whoever gives in to such self-absorbed vanity
has huge misery hiding inside them.”*

Pope Francis

Conteúdo

1	Introdução	1
1.1	Motivação	1
1.2	Caracterização do Problema	1
1.3	Estrutura da Dissertação	3
2	Estado da Arte	5
2.1	Introdução	5
2.2	Movimento coordenado de veículos autônomos, tendo restrições de comunicações	5
2.3	Redes de Sensores sem Fios (WSN)	6
2.4	Controlo de Sistemas em Rede	7
2.5	Filtros de Kalman	8
3	Comunicação em rede	9
3.1	Introdução	9
3.2	Modelos Probabilísticos	9
3.2.1	Modelo Uniforme	9
3.2.2	Modelo Linear	9
3.2.3	Modelo Exponencial	10
3.3	Valor Esperado	11
3.4	Resumo	12
4	Topologias de Rede	15
4.1	Introdução	15
4.1.1	Sistema UTM	15
4.2	Comunicação direta	15
4.3	Comunicação Fixa	16
4.4	Uso de Um Veículo <i>Bridge</i>	17
4.5		20
4.6	Conclusão	20
5	Validação de algoritmo	21
5.1	Introdução	21
5.2	Inicialização	21
5.3	Simulação	22
5.4	Resultados	22
5.5	Conclusão	23

6	Simulação e Resultados	25
6.1	Introdução	25
6.2	Resultados	25
6.3	Conclusão	25
7	Conclusões e Trabalho Futuro	27
A	Algoritmos	29
A.1	Classe Pos	29
A.2	Aquisição de dados	30
A.3	Line Intersect	30
A.4	Plot Arc	31
	Referências	33

Lista de Figuras

1.1	ASV Zarco	2
1.2	AUV MARES	2
1.3	AUV TRIMARES	3
3.1	Modelo uniforme	10
3.2	Modelo linear	11
3.3	Modelo exponencial	11
3.4	Valor esperado do modelo probabilístico linear.	13
3.5	Valor esperado do modelo probabilístico exponencial.	13
4.1	Disposição de N veiculos entre o emissor e o recetor no estado de repouso	16
4.2	Exemplo em que a linha que une o veículo <i>bridge</i> e o recetor intersecta o raio do emissor	18
4.3	Exemplo em que a linha que une o veículo <i>bridge</i> e o recetor não intersecta o raio do emissor	19
5.1	Resultado da simulação para os diferentes casos	23
6.1	Resultado da simulação dos quatro algoritmos diferentes, onde varia a topologia da rede	26

Lista de Tabelas

6.1	Valores usados na comparação das topologias	25
-----	---	----

Abreviaturas e Símbolos

ASV	Autonomous Surface Vehicle
AUV	Autonomous Underwater Vehicle
GPS	Global Positioning System
MARES	Modular Autonomous Robot for Environmental Sampling
SSH	Secure SHell
UTM	Universe Transverse Mercator
WSN	Wireless Sensor Network (Rede de Sensores sem Fios)

Capítulo 1

Introdução

1.1 Motivação

Quando em missão, é, de todo, importante que não haja quaisquer falhas na comunicação entre os vários intervenientes. Ao falar em veículos autónomos, tem-se, como preocupação, não só a detecção destas falhas, mas também a sua correcção.

Uma maneira de minimizar estas falhas, é usar veículos secundários para preencher lacunas, realizando pontes entre o veículo emissor e o recetor.

1.2 Caracterização do Problema

O problema a tratar é o desenvolvimento de estratégias de controlo coordenado e de localização distribuída para uma rede de veículos aquáticos, entrando em consideração com as restrições da rede de comunicação. Este problema divide-se em 3 sub-problemas:

- A formação adotada pelos veículos, uns com os outros;
- Otimização da comunicação entre eles de modo a que esta seja o mais eficiente possível;
- Junção de localização subaquática cooperativa com recurso a uma mistura de redes acústicas com redes sem fios.

As restrições impostas pela rede de comunicação são as seguintes:

- Intermitências inerentes a comunicações sem fios, sendo mais frequentes com o tipo de plataforma a ser usada, pois a ondulação da água pode limitar o alcance da comunicação;
- O próprio alcance referido no ponto anterior é um dos desafios a enfrentar, uma vez que é possível haver casos em que o alcance necessário seja relativamente grande;
- O atraso das comunicações que influenciam a instabilidade o sistema, como será referido no capítulo [2](#).

Para além dos desenvolvimentos conceptuais, os algoritmos estudados serão aplicados nos veículos aquáticos presentes na FEUP.

Os veículos de superfície autónomos (ASV), Zarco e Gama, fig 1.1, têm a forma de um catamarã, são altamente modulares e foram desenhados para operar em águas calmas. Contém um computador a bordo de modo a executar as missões de forma completamente autónoma ou controlada remotamente e onde pode armazenar os dados internos e da carga. Também possui uma ligação wifi com a qual pode comunicar com uma estação em terra. Tem 1,5m de comprimento, 1m de largura e pesa cerca de 50kg, podendo aguentar cargas de até 20kg e consegue atingir 3nós de velocidade, o que corresponde a 5,556 km/h.

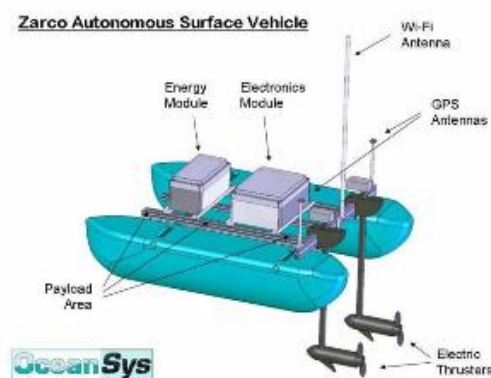


Figura 1.1: ASV Zarco

O MARES, fig 1.2, é um Veículo subaquático autónomo (AUV), também altamente modular, foi desenhado para operações a baixas profundidades. A sua configuração permite que lhe sejam acoplados uma grande variedade de sensores oceanográficos e tem um de sensores de navegação para garantir que se mantém na trajetória planeada. Tem forma cilíndrica com 20cm de diâmetro e 1,5m de comprimento e pesa cerca de 32kg, podendo atingir profundidades na ordem dos 100m e velocidades de cerca de 2m/s.



Figura 1.2: AUV MARES

O TriMARES, fig 1.3 é um AUV composto por 3 partes baseadas no MARES montadas em forma de triângulo. Foi desenhado para ter uma maior capacidade de carga, de modo a poder transportar, por exemplo, câmaras de alta qualidade e sonares. Os seus sensores de navegação incluem um IMU, um sensor de pressão e um sistema de posicionamento acústico. Mede 1,5m de

comprimento, 80cm de largura composta e 50 cm de altura composta e pesa 75kg. Pode atingir os 100m de profundidade e uma velocidade de 2m/s.

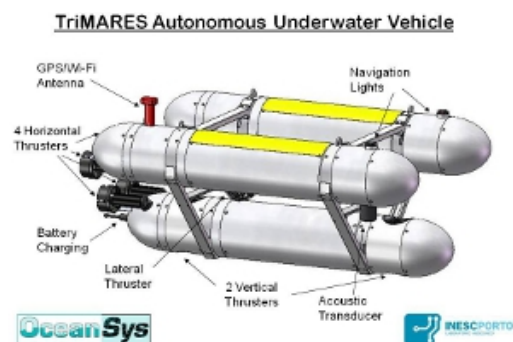


Figura 1.3: AUV TRIMARES

1.3 Estrutura da Dissertação

Este documento está dividido em seis partes:

No capítulo 2 é revisto o estado da arte que refere o movimento coordenado de veículos autônomos com restrições de comunicações. É, também, introduzido o conceito de redes de sensores sem fios e o controlo de sistemas em rede, assim como filtros de Kalman como ferramenta de controlo.

No capítulo 3 são estudados modelos probabilísticos de comunicação, nomeadamente, um modelo uniforme, um modelo linear e um modelo exponencial.

No capítulo 4 são estudadas 4 topologias distintas para a comunicação entre dois veículos.

Por fim, no capítulo 6 são comparados quatro algoritmos diferentes em termos de tempo médio usado para estabelecer uma comunicação, desde que, estando os veículos em repouso, partem para estabelecer a comunicação e retornam à posição original.

Capítulo 2

Estado da Arte

2.1 Introdução

Neste capítulo é apresentado o estado da arte. É, primeiramente, apresentado o movimento coordenado de veículos em rede, seguido da modelização de redes sem fios e dos problemas levantados pelo controlo em rede e uma possível solução através da utilização de filtros de kalman.

2.2 Movimento coordenado de veículos autónomos, tendo restrições de comunicações

Uma rede de comunicações pode ser modelada recorrendo ao uso de grafo onde os vértices são os nodos de comunicação e as arestas ligam os nodos que conseguem comunicar. Um grafo diz-se conectado quando dois vértices quaisquer podem ser unidos por um caminho de tamanho arbitrário. [1]

Matematicamente, um grafo pode ser representado pela matriz de incidência M de dimensão $n \times \varepsilon$, sendo n o número de vértices e ε o número de arestas. Esta matriz é preenchida da seguinte forma:

$$m_{kl} = \begin{cases} +1 & \text{se o vértice } k \text{ for o início da aresta } l \\ -1 & \text{se o vértice } k \text{ for o fim da aresta } l \\ 0 & \text{nos outros casos} \end{cases} \quad (2.1)$$

O laplaciano de um grafo, L , é definido como sendo $L = MM^T$ e tem ordem $n \times n$. Por construção, L é independente da orientação dada a cada aresta na matriz M , os seus valores próprios são não negativos e o vetor 1_n pertence ao seu núcleo, isto é, $L1_n = 0_n$. No caso de grafos conectados, L tem um valor próprio em zero e vetor próprio 1_n .

De seguida, são apresentadas considerações sobre redes de comunicações, supondo topologias fixa, onde o grafo é fixo ao longo do tempo, e topologias variáveis, onde a conectividade do grafo pode variar.

No primeiro caso, podem-se considerar restrições nas comunicações bidirecionais e unidirecionais.

Nas comunicações bidirecionais, assumem-se duas condições:

- Se o nodo i comunicar com o nodo j , o nodo j também consegue comunicar com o nodo i , isto é, se N_k for o conjunto de nodos com os quais o nodo k consegue comunicar, então, $i \in N_j \Leftrightarrow j \in N_i$;
- O grafo está conectado. Caso contrário, haveria dois ou mais conjuntos de nodos incapazes de comunicar uns com os outros.

Nas comunicações unidirecionais, a primeira condição das comunicações bidirecionais, já não se aplica. Apesar deste tipo de comunicações ser mais restritivo, é possível, na mesma, possível a execução de coordenação entre veículos e de seguimento de trajetórias lineares, circulares ou trajetórias paralelas de caminhos arbitrários.

Quanto às topologias de comunicação variáveis, tem-se breves perdas de conectividade e topologia "conectado em média".

Na primeira, considera-se que o grafo está alternadamente conectado e desconectado. Para isto, considera-se um conjunto com $2^{\bar{n}}$ grafos, correspondentes às possibilidades de conectividade do grafo, onde $\bar{n} = \frac{n \times (n-1)}{2}$ é o número de vértices e n é o número de nós do grafo. Deste conjunto, em cada interação, seleciona-se aleatoriamente um grafo e, daí, determina-se, de forma probabilística se o grafo está conectado ou não.

Na segunda topologia, considera-se a hipótese de o grafo nunca estar conectado, qualquer que seja o instante de tempo. No entanto, assume-se que, num intervalo de tempo finito, a união das várias posições do grafo, origina um grafo conectado. Isto é um grafo está conectado em média se, para todos os instantes de tempo $t \geq 0$ existir um $T > 0$ finito, tal que, a união dos grafos no intervalo $[t, t+T]$ seja um grafo conectado. Neste caso, a soma dos laplacianos dos grafos do intervalo, terá um valor próprio na origem, e vetor próprio 1_n .

2.3 Redes de Sensores sem Fios (WSN)

Uma WSN pode ser modelada como um conjunto de nodos numerados de 1 até N e uma matriz de conectividade C onde cada elemento $c_{ij} \in [0, 1]$ é a probabilidade do nodo j comunicar com o nodo i . Como a rede é aproximadamente simétrica, pode-se assumir que $C = C^T$ e $c_{ii} = 1, \forall i$. [2]

O grafo de conectividade- c $G_c = (N, E_c)$ está associado à matriz C como sendo o grafo onde (i, j) pertence ao conjunto das arestas se $c_{ij} \geq c$.

Existem três tipos de comunicação mais frequentemente usados:

- Broadcast – Um nó transmite uma mensagem para toda a vizinhança;
- Assymetric Gossip – Um nó transmite uma mensagem para um nó específico, pertencente à sua vizinhança;

- Symetric Gossip – Igual ao assymetric gossip, mas o nó que transmite fica à espera de resposta por parte do recetor.

2.4 Controlo de Sistemas em Rede

Num sistema de controlo em rede, os atuadores e os sensores comunicam com o controlador com recurso a uma rede de comunicação. O tempo que a informação demora entre as partes depende da topologia da rede e pode ser constante, tendo, no entanto, na maioria das vezes, variações aleatórias. [3]

Nestes sistemas, os atrasos advêm de três acções:

- Atraso de comunicação entre os sensores e o controlador;
- Atraso na computação dos dados no controlador;
- Atraso na comunicação entre o controlador e os sensores.

Sendo estes atrasos estocásticos, o sistema passa a ser variante no tempo, pelo que as técnicas usadas para sistemas invariantes no tempo não o possam ser diretamente utilizadas na análise deste sistema. Uma possível solução para este problema, será a utilização de buffers na realimentação, tornando, assim, os atrasos determinísticos e o sistema invariante no tempo.

Esta resolução levanta, no entanto, um outro problema: a informação a que o controlador, por vezes, tem acesso, é informação que já não é válida e isso leva a uma degradação da estabilidade e da eficiência do sistema. Assim sendo, torna-se, por vezes, mais comportável o cálculo de um controlador para o sistema invariante no tempo.

De seguida, são apresentados três possíveis modelos para a rede de comunicação:

- Rede modelada com atrasos constantes – Este é o modelo mais simples para a rede e é admissível até com redes com atrasos variáveis se a escala temporal do processo for muito superior ao atraso das comunicações. Neste caso, é usual usar o valor médio ou o pior caso. Nos casos em que isso não acontece, podem ser tiradas conclusões erradas quanto à estabilidade e eficiência do sistema.
- Rede modelada com atrasos independentes – Neste caso, os atrasos podem tomar várias formas como, por exemplo, o tempo de espera para que outras mensagens de maior prioridade sejam enviadas ou tempos de retransmissões. Como as ações do sistema não estão sincronizados, estes atrasos são estocásticos, e podem ser modelados com recurso a distribuições probabilísticas.
- Rede modelada com recurso a cadeias de Markov – Apesar de, com o modelo anterior, se assumir que os atrasos são independentes dos atrasos anteriores, isto nem sempre é verdade, pelo que é necessário que o sistema tenha memória, de modo a ser possível, correlacionar o atraso com os anteriores. Isto é possível usando cadeias de Markov que efetuam uma transição a cada comunicação feita.

2.5 Filtros de Kalman

Quando um robot se desloca num espaço, ele vai estimando a sua posição medindo valores internos, como velocidades, acelerações rotações das rodas. Estas medidas estão sempre afetadas de incertezas, as quais, vão sendo integradas ao longo do tempo. Na tentativa de reduzir estas incertezas, podem-se usar várias técnicas, uma das quais é o filtro de Kalman. Este filtro usa informações do robot em relação ao mundo, como posições relativas a objetos conhecidos, posição GPS, orientação medida através de uma bússola, entre outras. [2]

No caso da existência de vários robots a deslocarem-se, cada um deles pode enviar as estimações da sua posição e da posição dos outros para cada um dos outros robots e receber as suas respetivas estimativas. Com estes dados, usando um filtro de Kalman, pode corrigir a sua estimativa e o seu erro relacionado. Este filtro é um filtro de Kalman distribuído.

Capítulo 3

Comunicação em rede

3.1 Introdução

Neste capítulo são apresentados modelos probabilísticos de comunicação em rede e um algoritmo para, dependendo da topologia de rede, medir a potência do sinal e, conseqüentemente, determinar o modelo correspondente.

3.2 Modelos Probabilísticos

Um modelo probabilístico é uma forma matemática de, dada uma variável aleatória X e um evento A , determinar a probabilidade de ocorrência desse evento.

Em seguida são apresentados três modelos diferentes referentes à comunicação entre dois veículos, onde p é a probabilidade de sucesso da comunicação numa dada tentativa, sendo a probabilidade de insucesso $1-p$.

3.2.1 Modelo Uniforme

O modelo uniforme tem uma probabilidade da seguinte forma, ilustrada na figura 3.1:

$$p = \begin{cases} 1 & \text{se } 0 \leq x \leq r \\ 0 & \text{se } x > r \end{cases} \quad (3.1)$$

3.2.2 Modelo Linear

No entanto, a probabilidade de comunicação quando $x \leq r$ não é 1 e verifica-se, também, que à medida que x tende para zero, esta probabilidade tende para um máximo M .

Neste caso, propõem-se dois modelos probabilísticos, um linear e um exponencial.

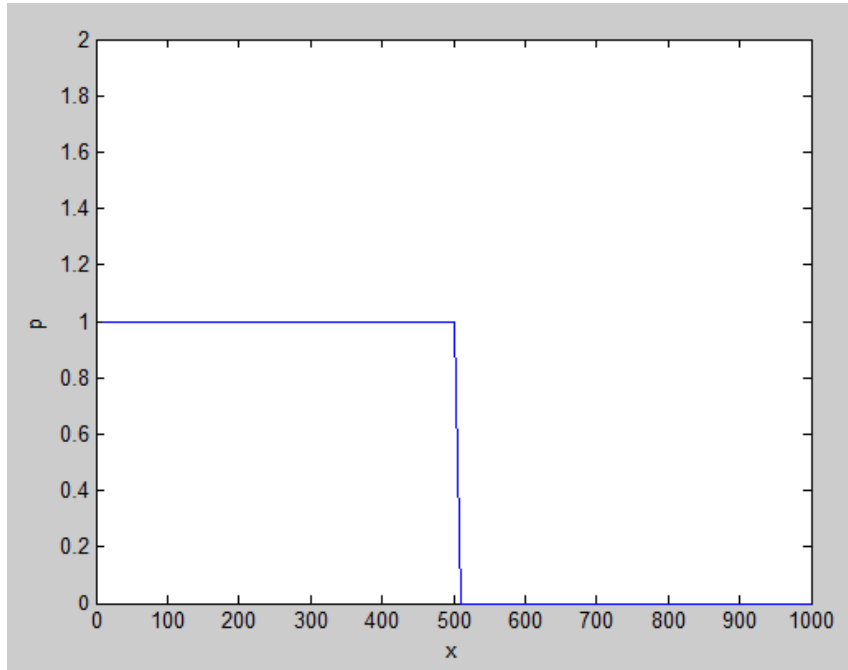


Figura 3.1: Modelo uniforme

No caso linear, tem-se a seguinte probabilidade:

$$p = \begin{cases} 1 & \text{se } 0 \leq x \leq r_{min} \\ \frac{x-r_{max}}{r_{min}-r_{max}} & \text{se } r_{min} < x \leq r_{max} \\ 0 & \text{se } x > r_{max} \end{cases} \quad (3.2)$$

onde k é o declive da recta e r_{min} e r_{max} são os raios entre os quais a probabilidade de sucesso de comunicação desce de 1 até 0. Este modelo encontra-se ilustrado na figura 3.2.

3.2.3 Modelo Exponencial

No caso exponencial, tem-se:

$$p = \begin{cases} 1 & \text{se } x \leq r \\ e^{-\lambda(x-r)} & \text{se } x > r \end{cases} \quad (3.3)$$

onde λ é o coeficiente de decaimento do sinal com a distância, visível na figura 3.3.

Estes modelos podem ser usados para modelizar retransmissões. Sendo p independente entre várias tentativas, tem-se a seguinte equação para n tentativas.

$$p_n = (1 - p)^{(n-1)} \times p \quad (3.4)$$

onde p_n é a probabilidade de ter uma transmissão bem sucedida na iteração n .

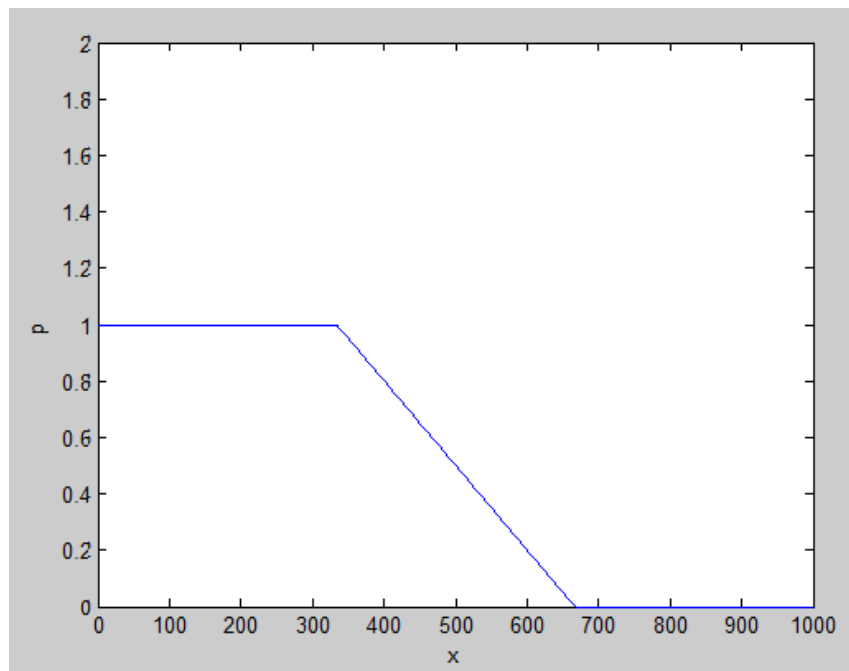


Figura 3.2: Modelo linear

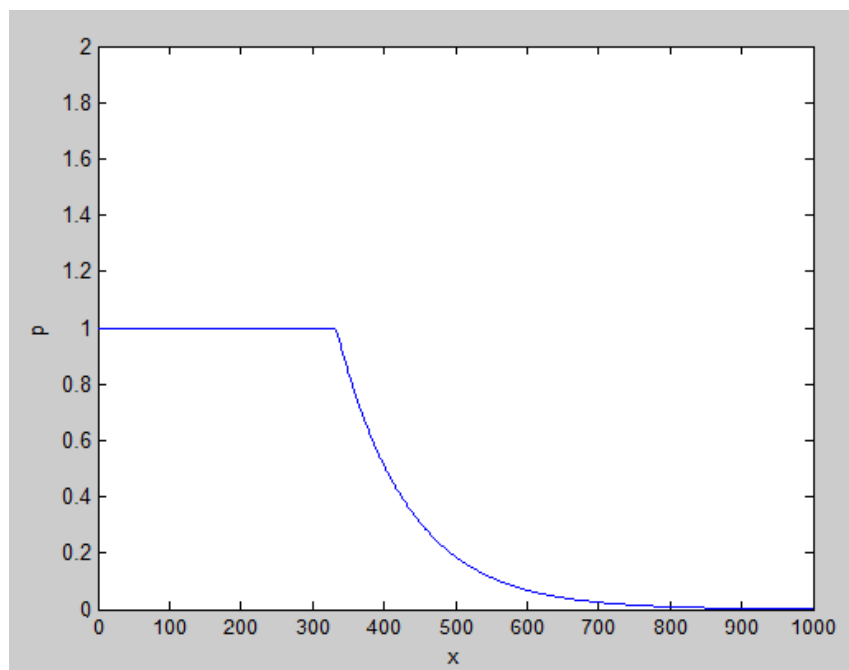


Figura 3.3: Modelo exponencial

3.3 Valor Esperado

Na aplicação destes modelos probabilísticos, deve-se usar a noção de valor esperado E , de modo a se ter melhor conhecimento dos modelos em si.

dada uma variável aleatória X , o valor esperado é dado pelas seguintes equações, para modelos discretos e contínuos, respetivamente:

$$E(X) = \sum_{i=1}^{\infty} x_i p(x_i) \quad (3.5)$$

$$E(X) = \int_{-\infty}^{\infty} x p(x) dx \quad (3.6)$$

os modelos estudados, cuja probabilidade é dada pela equação 3.4, são modelos contínuos e o seu valor esperado é:

$$E(X) = \frac{1}{p} \quad (3.7)$$

Este valor representa, nestes modelos, o número esperado de retransmissões de uma dada mensagem até esta ser enviada com sucesso.

Assim, o valor esperado, no caso uniforme, é 1 quando a probabilidade de transmissão é 1 e infinito quando a probabilidade é 0, como seria de esperar.

Para os modelos linear e exponencial, o valor esperado, em função da distância está ilustrado nas figuras 3.4 e 3.5 e segue as seguintes equações.

$$E(X_{linear}) = \begin{cases} 1 & \text{se } 0 \leq x \leq r_{min} \\ \frac{r_{min}-r_{max}}{x-r_{max}} & \text{se } r_{min} < x \leq r_{max} \\ \infty & \text{se } x > r_{max} \end{cases} \quad (3.8)$$

$$E(X_{exponencial}) = \begin{cases} 1 & \text{se } x \leq r \\ \frac{1}{e^{-\lambda(x-r)}} & \text{se } x > r \end{cases} \quad (3.9)$$

Como se pode ver, à medida que os veículos se vão distanciando, o número esperado de retransmissões aumenta de forma exponencial, tendendo para o infinito.

3.4 Resumo

Após o estudo destes diferentes modelos, é necessária a determinação do modelo e coeficientes relativos a cada topologia de rede. No apêndice A é apresentado o algoritmo A.2 para a recolha de dados relativos à potência de sinal entre dois veículos em função da distância entre os mesmos.

Para isso, após o programa a correr no veículo, denominado emissor, criar um cliente de SSH, liga-se ao *router* presente no mesmo e, em cada iteração, mede a potência e o ruído sinal entre o seu *router* e o do veículo com o qual está ligado, em diante denominado recetor .

Após essa acção, usando a classe *Pos* definida no algoritmo A.1, actualiza a sua posição e do recetor e calcula a distância entre eles, guardando toda a informação num ficheiro.

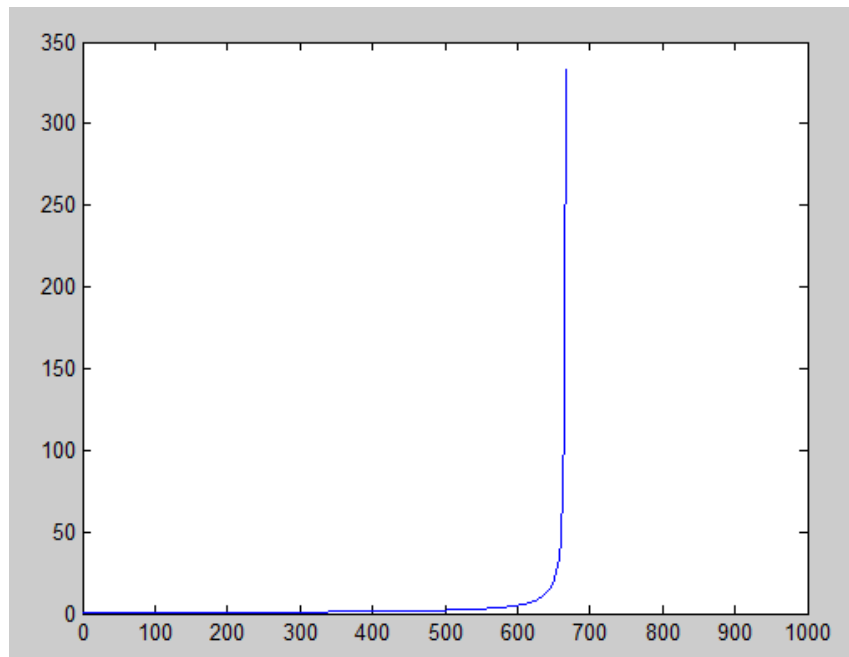


Figura 3.4: Valor esperado do modelo probabilístico linear.

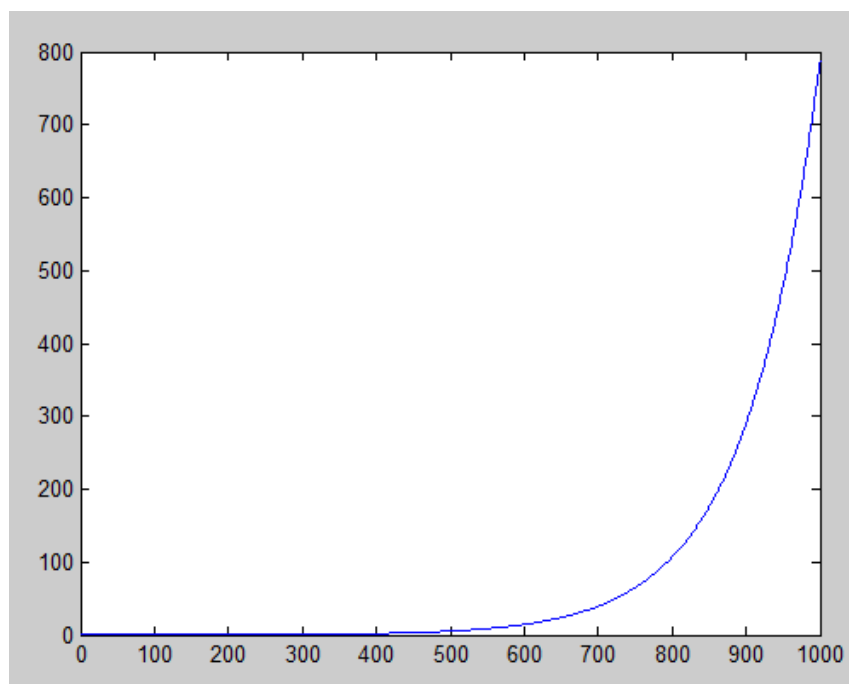


Figura 3.5: Valor esperado do modelo probabilístico exponencial.

Capítulo 4

Topologias de Rede

4.1 Introdução

Neste capítulo serão apresentados 4 topologias da rede as quais serão comparadas no capítulo 5.

Estas topologias, assim como os algoritmos a elas associados, foram estudadas no decorrer desta dissertação.

4.1.1 Sistema UTM

A primeira consideração a realizar, quando se fala em comunicação, é a distância entre os intervenientes. O sistema GPS, contendo medidas angulares torna este trabalho complicado. Nesse caso, tendo em consideração, que os veículos se encontram sempre numa distância muito inferior ao do raio da Terra, pode-se usar um outro sistema, o UTM.

Este sistema consiste na divisão da superfície da Terra numa grelha com 20x60 campos com 6º de largura, permitindo aproximar as coordenadas GPS dentro de cada campo a um plano cartesiano, permitindo medir distâncias entre dois pontos, assumindo-os num mesmo plano.

A conversão de GPS para UTM é feita através das fórmulas presentes em [11].

4.2 Comunicação direta

A primeira topologia consiste no emissor e no recetor deslocarem-se em direção um do outro, efectuarem a comunicação e voltarem ao local original.

A função que dá o tempo total da comunicação em função da distância a que se encontram os dois veículos é

$$t_{tot} = \frac{d - \min(r_{emissor}, r_{recetor})}{v} + \frac{t_{tr}}{p} \quad (4.1)$$

, onde t_{tot} é o tempo total da comunicação, d é a distância entre os veículos, $r_{emissor}$ e $r_{recetor}$ são os raios do emissor e do recetor, respectivamente, v é a velocidade dos veículos, t_{tr} é o tempo de transmissão da mensagem e p é a probabilidade de sucesso de uma dada tentativa de transmissão.

4.3 Comunicação Fixa

Na segunda topologia, é estudada a possibilidade de estarem N veículos entre o emissor e o recetor, de modo a diminuir o tempo total de comunicação.

Estes veículos estão, em repouso, distribuídos uniformemente entre o emissor e o recetor, de acordo com a figura 4.1.

Cada veículo desloca-se em direção ao seu emissor, seguindo para o seu recetor, voltando, por fim, ao local de repouso. Na figura 4.1, a variável x , corresponde ao deslocamento do recetor de um dado veículo, enquanto este se desloca até ao emissor. É fácil deduzir que a função que relaciona o tempo total de transmissão e o número de veículos é independente desta variável, uma vez que, o tempo será sempre o mesmo para um dado número de veículos intermedios, qualquer que seja o valor tomado por x .

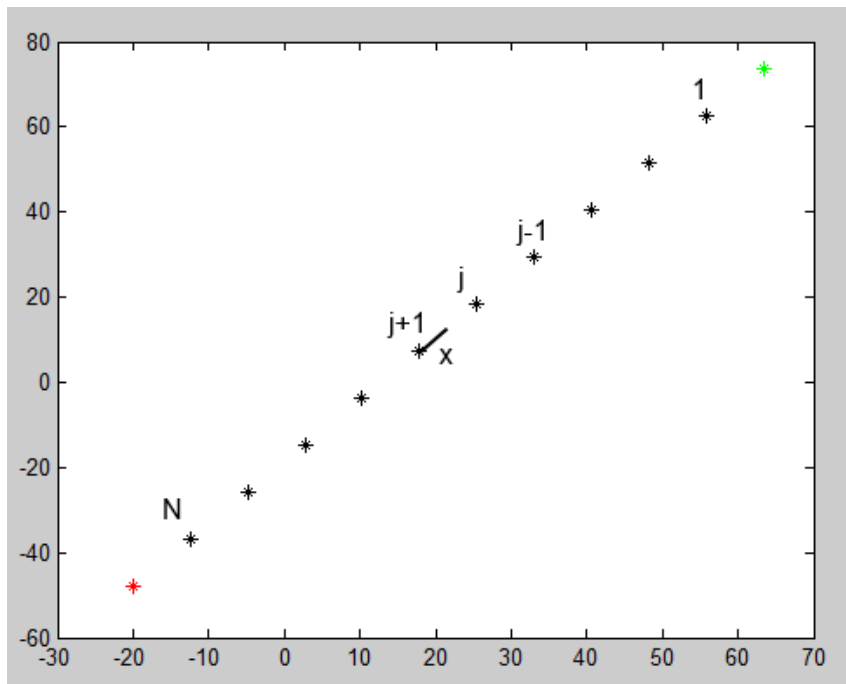


Figura 4.1: Disposição de N veículos entre o emissor e o recetor no estado de repouso. O emissor encontra-se representado por um ponto verde e o recetor por um ponto vermelho. Os veículos encontram-se igualmente espaçados entre eles de modo a minimizar o caminho a percorrer. Na iteração em causa, o veículo j e o seu emissor (o veículo $j - 1$) deslocam-se em direção um do outro para estabelecer a comunicação. Enquanto isso acontece, o seu recetor (o veículo $j + 1$) desloca-se na direção do veículo j por uma distância x .

Assim, a função que relaciona o tempo total de comunicação e o número de veículos é a seguinte.

$$t_{tot} = \frac{N+2}{N+1} \times \frac{d}{v} + \frac{t_{tr} \times (N+1)}{p} \quad (4.2)$$

sendo N o número de veículos entre o emissor e o recetor .

Sendo esta a função a minimizar, basta determinar-se a sua derivada e igualá-la a zero, obtendo-se a seguinte função:

$$N = \sqrt{\frac{d \times p}{t_{tr} \times v}} - 1 \quad (4.3)$$

Sendo esta uma função $f : \mathbb{R} \rightarrow \mathbb{R}$ e sendo N um número inteiro, torna-se necessário calcular o valor da equação 4.3 para $\lfloor N \rfloor$ e para $\lceil N \rceil$, escolhendo o valor de i que dê o menor resultado.

4.4 Uso de Um Veículo *Bridge*

Nesta topologia, assume-se uma rede distribuída de veículos onde o emissor e o recetor podem não estar dentro do alcance um do outro. Nesse caso, é escolhido um veículo livre para efectuar a ponte entre estes dois de modo a minimizar o tempo total de comunicação.

Estando os N veículos em missão, considera-se, em cada iteração com comunicação, 3 casos diferentes para os 2 veículos envolvidos:

1. estão ambos dentro do raio de comunicação;
2. os raios de comunicação intersectam-se;
3. os raios de comunicação não se intersectam.

No primeiro caso, ambos os veículos, estão automaticamente aptos para iniciar a comunicação. No segundo e terceiro casos, esta condição já não se verifica, sendo necessário um terceiro veículo para fazer a ponte, em diante denominado *bridge* .

No segundo caso, o veículo *bridge* deve-se colocar, na linha que une os 2 veículos no ponto médio da intersecção dos raios de comunicação, que representa o centro da intersecção, seleccionando-se para *bridge* , o veiculo que se encontrar mais próximo desse ponto.

No terceiro caso, é necessário que o veículo *bridge* se desloque, primeiro em direcção à área de cobertura do veículo emissor e, após receber a mensagem a transmitir, deslocar se para o recetor . Neste caso, é seleccionado para *bridge* que percorra o menor caminho neste processo.

Para determinar este caminho, existem 2 casos, dependendo do posicionamento entre os 3 veiculos, isto é, se a linha que une o veículo *bridge* e o recetor intersecta, ou não o raio de comunicação do veículo emissor.

Para se determinar esta situação, verifica-se se a distância entre o emissor e essa linha é igual ou inferior ao raio de comunicação. Em caso afirmativo, verifica-se que a intersecção ocorre.

Se intersectar, essa linha representa o caminho mais curto a percorrer pelo veículo. Nesse caso, ele desloca-se até ao ponto de intersecção do caminho com a sua perpendicular que passa pelo ponto onde o emissor se localiza para receber a mensagem e depois, deslocar-se até conseguir transmitir a mensagem para o recetor, como se pode verificar na figura 4.2.

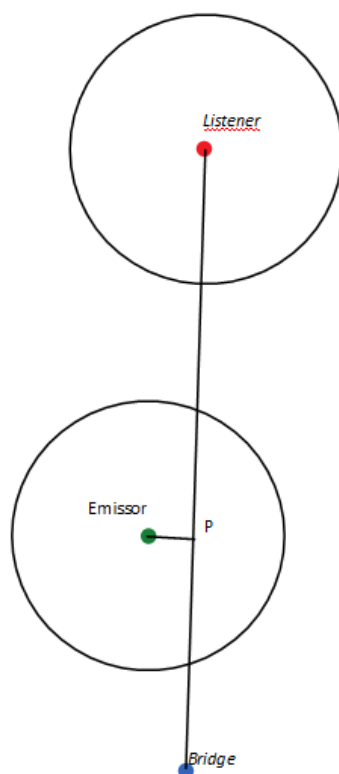


Figura 4.2: Exemplo em que a linha que une o veículo *bridge* e o recetor intersecta o raio do emissor

Em caso negativo, o veículo *bridge* terá de se deslocar até um ponto dentro do raio de comunicação do emissor e, após receber a mensagem, até ao recetor para a transmitir de tal modo que minimize o caminho percorrido, como visto na figura 4.3.

Neste caso, o caminho a percorrer será obtido da seguinte forma:

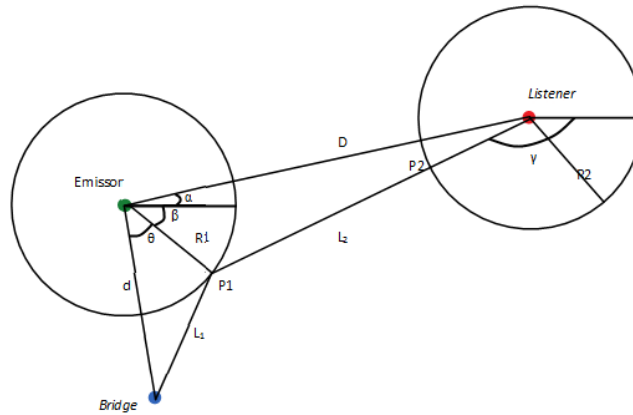


Figura 4.3: Exemplo em que a linha que une o veículo *bridge* e o recetor não intersecta o raio do emissor

$$d = \begin{cases} (1, 0) & \text{se } x_{recetor} < x_{bridge} \\ (-1, 0) & \text{nos outros casos} \end{cases} \quad (4.4)$$

$$D = (x_{emissor} - x_{recetor}, y_{emissor} - y_{recetor}) \quad (4.5)$$

$$\alpha = \arctan \left(\frac{D(1)}{D(0)} \right) \quad (4.6)$$

$$\beta = \arccos \left(\frac{d \cdot D}{||d|| ||D||} \right) - \theta - \alpha \quad (4.7)$$

$$P_1 = \begin{cases} (x_{bridge}, y_{bridge}) & \text{no primeiro caso} \\ (x_{emissor}, y_{emissor}) + (R_1 \cos(\beta), R_1 \sin(\beta)) & \text{no segundo caso} \end{cases} \quad (4.8)$$

$$T = P_1 - (x_{recetor}, y_{recetor}) \quad (4.9)$$

$$\gamma = \arctan \left(\frac{T(1)}{T(0)} \right) \quad (4.10)$$

$$P_2 = (x_{recetor}, y_{recetor}) + (R_1 \cos(\gamma), R_1 \sin(\gamma)) \quad (4.11)$$

$$L_1 = \sqrt{(x_{bridge} - P_1(0))^2 + (y_{bridge} - P_1(1))^2} \quad (4.12)$$

$$L_2 = \sqrt{(P_2(0) - P_1(0))^2 + (P_2(1) - P_1(1))^2} \quad (4.13)$$

$$L = L_1 + L_2 \quad (4.14)$$

Sendo todos os valores, excepto θ e β constantes e β dependente de θ através da equação (4.7), este torna-se num problema de optimização, tendo, como variável, o ângulo θ .

4.5

A quarta topologia é uma mistura entre a primeira e a terceira topologias de modo a otimizá-las.

Em cada situação, é comparado o tempo total de comunicação em cada um desses algoritmos e é escolhida a ação em conformidade com o resultado.

4.6 Conclusão

Neste capítulo foram apresentadas 4 topologias distintas para a comunicação dentro de uma rede. No capítulo 5 será validado o terceiro algoritmo e no capítulo ?? as 4 topologias serão comparadas através de um ambiente de simulação que foi desenvolvido para o efeito.

Capítulo 5

Validação de algoritmo

5.1 Introdução

Neste capítulo será validado o algoritmo relativo à terceira topologia apresentada no capítulo 4;

Relembrando, quando dois veículos pretendem comunicar, caso não estejam dentro do raio de comunicação, é necessária a utilização de um veículo *bridge*.

Nesse caso, dependendo da distância entre o emissor e o recetor, o veículo *bridge* terá de se deslocar para um ponto, se os raios de comunicação dos dois primeiros se intersectarem, ou dois, se não.

Caso se intersectem, o veículo *bridge* deve-se deslocar para o centro da intersecção. Se não, dois casos são possíveis de considerar.

- A linha que une o veículo *bridge* e o recetor intersectam o raio de comunicação do emissor
- A linha que une o veículo *bridge* e o recetor não intersectam o raio de comunicação do emissor

Os dois pontos pelos quais o veículo *bridge* deve passar são dados pelas equações (4.4) a (4.11).

Em qualquer dos casos, é seleccionado, de entre os outros $N-2$ veículos, o que minimize a distância percorrida.

No caso de intersecção entre os raios de comunicação do emissor e do recetor, esta distância percorrida é a distância entre o veículo *bridge* e o ponto para onde se deve deslocar. No outro caso, é calculada pelas equações (4.12) a (4.14).

5.2 Inicialização

Para efeitos de simulação, os veículos foram modelizados pela sua posição (x,y) e pelo seu raio de comunicação. Existe, também uma área onde os barcos se podem movimentar que tem a

forma quadrada e que é modelizada pelo tamanho do lado. O último modelo usado, é a ordem de comunicações que consiste numa matriz de tamanho $N \times N$ onde, na posição (i,j) contém o número de iterações que faltam até o veículo i comunicar com o veículo j .

A inicialização da simulação é efectuada uma colocação aleatória dos N veículos dentro da área e da matriz de comunicações. Esta colocação segue um modelo probabilístico uniforme entre os limites do mundo. O raio de comunicação também segue um modelo uniforme entre 25m e 50m.

Durante esta inicialização é garantido, por motivos de simplificar, que dois barcos não tentam comunicar em simultâneo. Isto é alcançado através de uma matriz cujo índice ij é único e representa o número de iterações restantes para o veículo i comunicar com o veículo j .

5.3 Simulação

Em cada iteração, é executada uma deslocação aleatória dos N veículos, garantindo que se mantém dentro da área disponível, seguido da determinação da comunicação a ser executada nessa iteração. Caso, não exista, passa-se à iteração seguinte.

Determinando-se qual o emissor e o recetor da comunicação, armazenam-se os dados relativos a cada, nomeadamente, posição e raio de comunicação, e determina-se a intersecção dos mesmos, caso exista.

Após esta acção, determina-se qual dos 3 casos enumerados no início da secção ?? e aplica-se o algoritmo descrito na mesma secção dependendo do caso.

Por fim, é mostrado o resultado da iteração num gráfico com as seguintes características:

- O emissor é apresentado com um ponto verde;
- O recetor é apresentado com um ponto vermelho;
- Todos os outros veículos são apresentados com pontos pretos;
- O recetor é apresentado com um ponto vermelho;
- Os raios de comunicação do emissor e do recetor são apresentados por circunferências azuis;
- No caso de intersecção dos raios, esta é apresentada a vermelho sobre as circunferências azuis;
- Caso seja necessário um veículo *bridge*, o caminho percorrido é apresentado com uma linha azul e o ponto onde termina é apresentado por um ponto azul.

5.4 Resultados

Da simulação obtiveram-se os resultados pretendidos, sendo que, quando necessário, era seleccionado um veículo para *bridge*, o qual se dirigia ao emissor, se não estivesse dentro do range e, posteriormente, ao recetor.

Para os resultados ilustrados, seguidamente nas figura 5.1, foi simulada a situação de 10 veículos numa área quadrada de 1000 pontos de lado.

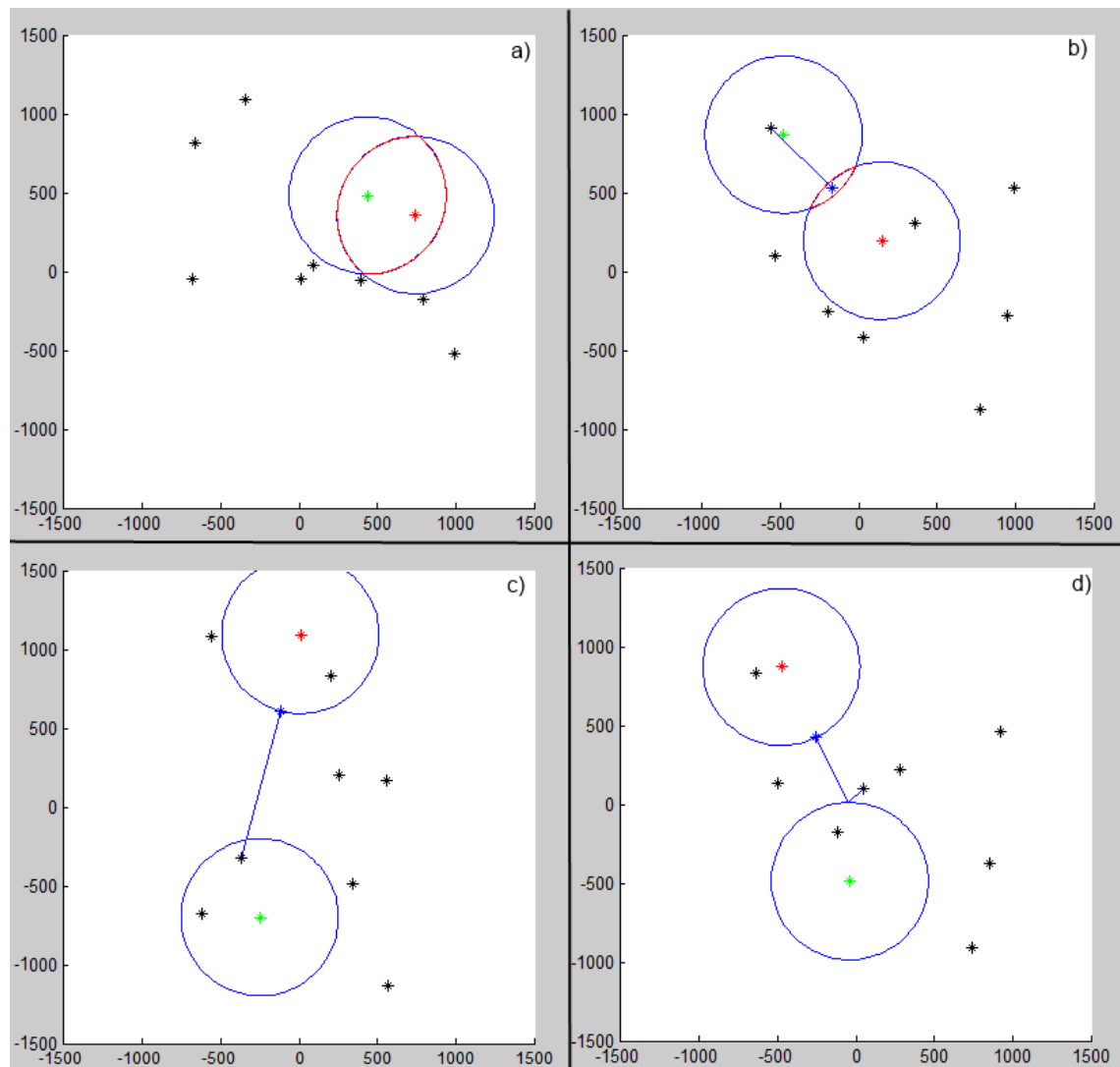


Figura 5.1: Resultado da simulação para os diferentes casos: a) Ambos os veículos encontram-se dentro do raio de comunicação, b) os raios de comunicação intersectam-se, sendo necessário um veículo *bridge* para efectuar a ponte, c) os raios de comunicação não se intersectam, estando, no entanto, o veículo *bridge* ao alcance do emissor, d) os raios de comunicação não se cruzam e é necessário o veículo *bridge* deslocar-se para o emissor antes de receber a mensagem

5.5 Conclusão

Neste capítulo, foi apresentada a simulação da aplicação do algoritmo descrito no capítulo 4. Para a elaboração da simulação, foi usada uma função adaptada do código por Matt Fig, para desenhar a intersecção entre as circunferências e uma função escrita por Paulo Silva para determinar a intersecção entre dois segmentos de recta.

Os algoritmos para ambas as funções estão no apêndice [A](#), respectivamente algoritmo [A.4](#) e [A.3](#).

Capítulo 6

Simulação e Resultados

6.1 Introdução

Neste capítulo serão comparadas as topologias apresentadas no capítulo 4. Para isto, foi desenvolvido um sistema de simulação de Monte Carlo que, em cada iteração, coloca 3 veículos no mundo, em posições e com raios aleatórios, do mesmo modo referido no capítulo 5.

A tabela 6.1 contém as constantes usadas nesta simulação.

Tabela 6.1: Valores usados na comparação das topologias

Velocidade do Barco	1,54 m/s
Tempo de Transmissão	0,02 s
Probabilidade de Sucesso	0.5

6.2 Resultados

Como se pode ver na figura 6.1, o algoritmo com o menor tempo de comunicação é o segundo. Este apresenta o valor médio mais baixo, sendo que, em todas as iterações, é calculado o tempo de cada algoritmo para um mesmo posicionamento dos vários veículo.

Entre os outros três, a diferença é bastante pequena sendo que se nota que a combinação do primeiro e do terceiro algoritmos levam a uma melhor *performance*, como era esperado.

Os resultados apresentam um desvio padrão elevado, devido à grande variação da distância entre o emissor e o recetor, que podem variar entre aproximadamente 0m e 283m

6.3 Conclusão

Como se pode verificar, o quarto algoritmo otimiza tanto o primeiro como o terceiro, pelo que se pode perceber que há situações em que o primeiro algoritmo é melhor que o terceiro e outras em que se verifica o oposto. Quanto ao segundo algoritmo, este tem um tempo médio maior que os outros 3, embora semelhante, pelo que, no caso em que haja um emissor e um recetor fixos

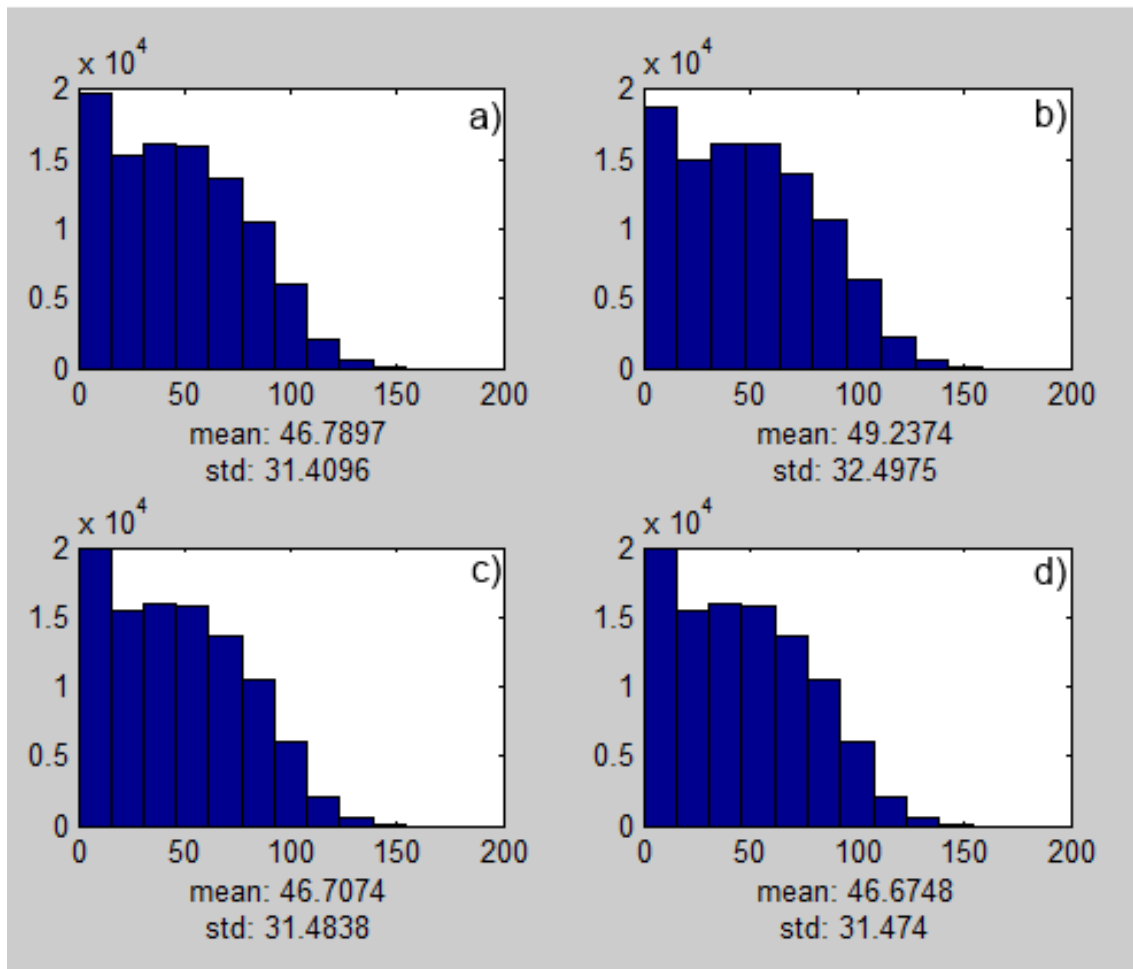


Figura 6.1: Resultado da simulação dos quatro algoritmos diferentes, onde varia a topologia da rede: a) Primeiro algoritmo - dois veículos (emissor e recetor), b) Segundo algoritmo - dois veículos (emissor e recetor) e N veículos *bridge* uniformemente distribuídos entre os dois, c) Terceiro algoritmo - dois veículos (emissor e recetor) e veículo *bridge* livre, d) Quarto algoritmo - combinação entre o primeiro e o terceiro algoritmos.

a uma distância aproximadamente fixa, se pode usar este algoritmo, com o objectivo de diminuir o tempo de computação e o espaço de memória utilizado.

Capítulo 7

Conclusões e Trabalho Futuro

Ao longo deste documento foram delineadas estratégias para que N veículos em missão consigam comunicar sempre que necessário, mesmo que não seja possível deslocarem-se até conseguirem estabelecer a comunicação com o veículo desejado.

Os algoritmos apresentados, foram aplicados em simulação, tendo sido verificados os comportamentos esperados por parte da movimentação dos veículos dentro da rede, aquando da necessidade de introdução de um veículo *bridge*.

O trabalho aqui desenvolvido tem aplicabilidade em qualquer situação que envolva comunicação entre dois agentes que podem não estar suficientemente próximos para estabelecer contacto, assumindo que as deslocações se desenvolvem em duas dimensões.

De futuro, pode-se considerar a utilização de veículos aquáticos, tanto para obtenção dos modelos probabilísticos de comunicação, como para aplicação do algoritmo desenvolvido no decorrer deste trabalho, assim como para se aplicar os algoritmos, tendo em consideração erros de posicionamento. Pode-se, também, considerar a aplicação de diferentes redes (Wireless, Radio Frequência, Acústica Aérea e Subaquática), permitindo, assim, a utilização de vários tipos de veículos, incluindo, barcos, submarinos e, possivelmente veículos aéreos, como helicópteros e aviões, adaptando-se, para isso, os algoritmos para três dimensões.

Anexo A

Algoritmos

Neste anexo, estão os algoritmos descritos durante o documento, apresentados em pseudo-código.

A.1 Classe Pos

```
class Pos(){
    initialize(ip = '127.0.0.1', port = 15000, lat = None, lon = None, alt = None){
        self.ip = ip
        self.port = port
        self.var =
        if lat is not None self.lat = lat
        if lon is not None self.lon = lon
        if alt is not None self.alt = alt }
    convertFromUtmToGps(){
        self.lat, self.lon = utm.to_latlon(self.x, self.y, self.n, self.s) }
    convertFromGpsToUtm(){
        self.x, self.y, self.n, self.s = utm.from_latlon(self.lat, self.lon) }
    getDistance(other){
        self.fromGPS()
        other.fromGPS()
        return sqrt((self.x-other.x)**2 + (self.y-other.y)**2 + (self.var['alt']-other.var['alt'])**2)
    }
}
```

A.2 Aquisição de dados

```

client = SSHClient()
client.connect(router's ip, username, password)
ip = listener vehicle's ip
while not KeyboardInterrupt {
    value = client.exec_command(command to read power and noise in wireless connection)
    pos = Pos(port listener)
    pos.updateGPSCoord()
    lat, lon, alt = getPosOther(ip)
    other = Pos(lat = lat, lon = lon, alt = alt)
    distance = pos.getDistance(other)
    t = timestamp()
    writeToFile(timestamp, distance, power and noise)
    delay(1 sec) }

```

A.3 Line Intersect

```

lineintersect(l1,l2){
//l1 and l2 are segments of line of the form [P1X, P1Y, P2X, P2Y]
m1 = slope of line 1
m2 = slope of line 2
b1 = y-intercept of line 1
b2 = y-intercept of line 2
b = [b1, b2]
a = [1 -m1; 1 -m2]
Pint = transpose of b * inverse of a
//find maximum and minimum values for the final test
l1minX=min([l1(1) l1(3)])
l2minX=min([l2(1) l2(3)])
l1minY=min([l1(2) l1(4)])
l2minY=min([l2(2) l2(4)])
l1maxX=max([l1(1) l1(3)])
l2maxX=max([l2(1) l2(3)])
l1maxY=max([l1(2) l1(4)])
l2maxY=max([l2(2) l2(4)])
//Test if the intersection is a point from the two lines because

```

```
//all the performed calculations were for infinite lines
if ((x<l1minX) | (x>l1maxX) | (y<l1minY) | (y>l1maxY) | (x<l2minX) | (x>l2maxX) |
(y<l2minY) | (y>l2maxY) )
  return None
else
  Return Pint
}
```

A.4 Plot Arc

```
plot_arc(th1,th2,xc,yc,r){
//th1 and th2 are the start and end of the arc in radians
//xc and yc are the center of the arc
//r is the radius of the arc
t = a row vector of 100 linearly equally spaced points between 0 and  $2\pi$ 
x = r*cos(t) + xc;
y = r*sin(t) + yc;
P = draw(x,y,'r'); }
```


Referências

- [1] Reza Ghabcheloo. Coordinated Path Following of Multiple Autonomous Vehicles. Tese de doutoramento, Instituto Superior Tecnico de Lisboa, April 2007.
- [2] Gianluigi Pillonetto e Sandro Zampieri. Cooperative control and estimation algorithms. Relatório técnico, University of Padova, August 2008.
- [3] Johan Nilsson. Real-time control systems with delays. Tese de mestrado, Lund Institute of Technology, 1998.
- [4] Nuno A. Cruz et al. Implementation of an underwater acoustic network using multiple heterogeneous vehicles. 2012.
- [5] Paul L. Meyer. Probabilidade - Aplicacoes a Estatistica. Livros Tecnicos e Cientificos Editora, 2.a edição, 1987.
- [6] Andre Sa. Modelacao e controlo dinamico de uma embarcacao autonoma. Tese de mestrado, Faculdade de Engenharia da Universidade do Porto, Julho 2009.
- [7] Antonio Santos. Modelo de comunicacao OceanSys. OceanSys.
- [8] TRIMARES Software de bordo.
- [9] OceanSys. Technology – Systems and Prototypes. [Online]. [2011?] [Citado 2013 Fev 13]; Disponível em: URL:<http://oceansys.fe.up.pt/technology.php>
- [10] Bureau International des Poids et Mesures. The International System of Units (SI). [Online]. 2006 [citado 2013 Fev 13]; Disponível em: URL:http://www.bipm.org/utis/common/pdf/si_brochure_8_en.pdf
- [11] Universal Transverse Mercator coordinate system. [Online]. [Citado 2013 Jun 17]; Disponível em: URL:http://en.wikipedia.org/wiki/Universal_Transverse_Mercator_coordinate_system