



Universidade de Aveiro
2012

Departamento de Engenharia Mecânica

**Eugénio Paulo
Andrade da Costa**

**Navegação em Ambientes Exteriores do Robô da
Série Atlas 2000**



**Eugénio Paulo
Andrade da Costa**

**Navegação em Ambientes Exteriores do Robô da
Série Atlas 2000**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Mecânica, realizada sob a orientação científica do Doutor Vítor Manuel Ferreira dos Santos, Professor Associado do Departamento de Engenharia Mecânica da Universidade de Aveiro.

O júri

Presidente

Professor Doutor Jorge Augusto Fernandes Ferreira
Professor Associado da Universidade de Aveiro

Professor Doutor Manuel Bernardo Salvador Cunha
Professor Associado da Universidade de Aveiro

Professor Doutor Vítor Manuel Ferreira dos Santos
Professor Associado da Universidade de Aveiro

Dedicatória

Dedico este trabalho aos meus pais.

Agradecimentos

Ao Professor Doutor Vítor Santos, quero agradecer por todo o conhecimento e sentido de responsabilidade que me transmitiu, por toda a disponibilidade e orientação que me facultou e pelo ambiente saudável que proporcionou em prol do sucesso deste trabalho.

A todos os membros do LAR e da equipa Atlas pela partilha de conhecimentos e boa disposição.

Ao Engenheiro António Festas pela disponibilidade e ajuda no fabrico das componentes mecânicas e ao Sr. Júlio do Departamento de Física pela conceção das placas de circuito impressas.

E claro, a todos que me acompanharam nos bons e maus momentos.

Palavras-chave

Grupo Atlas, Planeamento de trajetórias, Planeamento de caminhos, Sistema modular, CARMEN, Campos de Potencial, GPS, monitorização de baterias.

Resumo

Este trabalho propõe a implementação de um sistema de planeamento de caminhos para o meio exterior, através de GPS, no robô de série Atlas 2000. O projeto engloba as alterações mecânicas e de *software* necessárias para atingir este objetivo.

O robô de série Atlas 2000 pertence ao grupo de investigação e desenvolvimento Atlas, do Laboratório de Automação e Robótica (LAR) do Departamento de Engenharia Mecânica (DEM) da Universidade de Aveiro (UA), que se foca, entre outros, no estudo e desenvolvimento de sistemas de condução autónoma. Este robô participava anualmente no festival nacional de robótica, obtendo entre 2006 a 2011 boas classificações e prémios.

O trabalho iniciará pela migração do *software* existente no robô para uma arquitetura modular, em CARMEN, seguido pelas modificações mecânicas do robô para a sua deslocação em terrenos exteriores.

Após estas modificações será acoplado um recetor de GPS e concebido o *hardware* dedicado à sua alimentação e comunicação. Posteriormente será desenvolvido um algoritmo de planeamento de caminhos, com base no método de campos de potencial, em meios exteriores de modo autónoma e segura através de dados GPS. Será usado um sensor laser, existente no robô, para realizar a deteção de obstáculos.

Por fim ainda será implementado um sistema de monitorização das baterias do veículo.

Keywords

Atlas Group, Trajectories Planning, Road Planning, Modular System CARMEN, Potential Fields, GPS, Battery Monitoring.

Abstract

This thesis proposes the implementation of an outdoor path planning system, through GPS data, for the Atlas 2000 series robot. The project includes the mechanical and software changes necessary to achieve this goal.

The Atlas 2000 series robot belongs to the research and development team Atlas, of the Automation and Robotics Laboratory of the Mechanical Engineering Department at Aveiro University. This group focuses, among others, on the study and development of autonomous driving systems. This robot participated annually at the national robotics festival, achieving good classifications and prizes between 2006 and 2011.

The work begins by the migration of the existing software of the robot to a modular architecture, in Carmen, followed by the necessary mechanical alterations for the robot to move in an outdoor environment. After these modifications a GPS receptor will be installed and the dedicated hardware for its alimentation and communication will be produced. After all the hardware modifications, a path planning algorithm will be developed based on the potential field method using GPS data. The detection of obstacles will be insured by a laser sensor installed on the robot.

Finally, a battery monitoring system will also be installed on the robot.

Conteúdo

1.	Introdução.....	1
1.1.	Apresentação do Projeto Atlas	1
1.2.	O Festival Nacional de Robótica.....	3
1.3.	Objetivos/plano de intervenção	4
1.4.	Estrutura da dissertação	5
2.	Arquiteturas de <i>Software</i> e Navegação	7
2.1.	CARMEN/IPC	7
2.2.	GPS	12
2.3.	Planeamento de caminhos.....	14
2.3.1.	Método de Decomposição em células.....	15
2.3.2.	Método de mapa de estradas (<i>roadmap</i>).....	17
2.3.3.	Método dos Campos de Potencial	19
2.3.4.	Método Escolhido	23
3.	Intervenções na Plataforma.....	25
3.1.	Mudança de rodas	25
3.2.	Instalação do recetor de GPS.....	28
3.2.1.	Placa para a alimentação e comunicação do GPS.....	29
3.3.	Monitorização das baterias.....	31
3.3.1.	Placa de circuito impresso para a monitorização das baterias.....	32
4.	Sistemas Desenvolvidos	35
4.1.	Migração de todo o <i>software</i> para a filosofia CARMEN/IPC.....	35
4.2.	Monitorização das baterias.....	42
4.3.	Navegação por GPS.....	43
4.3.1.	Comunicação com o recetor GPS.....	44
4.3.2.	GPS <i>Mapping</i>	45

4.3.3.	Obtenção da Orientação	46
4.3.4.	Navegação Inicial para a obtenção de uma orientação válida	47
4.3.5.	Planeamento de Caminhos	49
4.3.5.1.	Planeamento de Trajetórias por Campos de Potencial Matriciais.....	49
4.3.5.2.	Planeamento de Trajetórias por Campos de Potencial Radial.....	57
4.3.6.	Funcionamento Global.....	61
5.	Ensaio e Resultados	63
5.1.	Monitorização das baterias.....	63
5.2.	Planeamento por Campos de Potencial Matricial	64
5.3.	Planeamento por Campos Potencial Radial	76
6.	Conclusões e Trabalhos Futuros	89
6.1.	Conclusões	89
6.2.	Trabalhos Futuros	91
7.	Referências.....	93
8.	Anexos.....	95
8.1.	Anexo A	95
8.2.	Anexo B	96
8.3.	Anexo C	97
8.4.	Anexo D.....	98
8.5.	Anexo E	98

Lista de Figuras

Figura 1 - Atlas 2000	2
Figura 2 - Altas MV	2
Figura 3 - Atlas Car1	2
Figura 4 - Estrutura do Atlas I/II [4].....	3
Figura 5 - Chassis do Atlas II.....	3
Figura 6 - Estado da eletrónica pré-2010 [2]	3
Figura 7 - Estado da eletrónica pós-2010 [2]	3
Figura 8 - Pista da competição de condução autónoma [2]	4
Figura 9 - Troca de mensagem básica em IPC.....	8
Figura 10 - Exemplo de sobrecarga numa comunicação IPC	9
Figura 11 - IPC - Comunicação com pedido/resposta	9
Figura 12 – Técnica de pedido/resposta com <i>heartbeat</i>	10
Figura 13 – Método de partilha com memória partilhada	12
Figura 14 - GPS - Constelação de satélites	13
Figura 15 - Método de decomposição em células (exato).....	16
Figura 16 - Decomposição em células (aproximado)	16
Figura 17 - <i>Roadmap</i> - Grafo de Visibilidade	18
Figura 18 - <i>Roadmap</i> - Diagrama de Voronoi.....	18
Figura 19 - Campos de Potencial.....	20
Figura 20 - Campo de Potencial - Mínimo local	21
Figura 21 - Roda do Atlas-2010.....	25
Figura 22 - Roda do Atlas-2011.....	25
Figura 23 - Adaptador para a roda dianteira (produto final e desenho 3D)	26
Figura 24 - Sistema de fixação da roda traseira	26
Figura 25 - Adaptador para a roda traseira (produto final e desenho 3D)	27
Figura 26 - Atlas 2011 com as rodas novas	27
Figura 27 - Recetor GPS SiRF Star III.....	29
Figura 28 - Localização do recetor GPS no robô	29
Figura 29 - Suporte para o recetor GPS	29
Figura 30 - Regulador de tensão LM317	30
Figura 31 – Conversor de sinal Max232	30
Figura 32 - Placa GPS.....	30

Figura 33 - Microcontrolador PIC18F2455.....	32
Figura 34 - Divisor resistivo.....	32
Figura 35 - Esquema das entradas do microcontrolador.....	32
Figura 36 - Circuito da monitorização das baterias	33
Figura 37 - Placa BM.....	34
Figura 38 - Módulos criados para a condução autónoma <i>indoor</i>	36
Figura 39 - Módulo da Base	37
Figura 40 - Módulo das câmaras.....	38
Figura 42 - Módulo laser	38
Figura 42 - Módulo de controlo remoto	39
Figura 43 - Módulo reproduzidor de som	39
Figura 44 - Módulo do IPM	41
Figura 45 - Exemplo de fusão de imagens de duas câmaras para perspetiva <i>birdview</i>	41
Figura 46 - <i>Printscreen</i> da monitorização das baterias.....	43
Figura 47 - Módulo de monitorização das baterias	43
Figura 48 - Campos de Potencial – <i>printscreen</i> do programa realizado em Matlab.....	44
Figura 49 - Módulo GPS	45
Figura 50 - GPS <i>Mapping</i> - Exemplo.....	46
Figura 51 - Módulo Orientação.....	47
Figura 52 - Módulo de Navegação Inicial.....	48
Figura 53 - Diagrama de funcionamento do módulo de Navegação Inicial.....	49
Figura 54 - Módulo de Planeamento de campos de potencial matricial.....	50
Figura 55 - Planeamento matricial - exemplo de carregamento e discretização do mapa	50
Figura 56 - Planeamento matricial - exemplo de colocação de obstáculos, destino, e posição atual .	51
Figura 57 - Planeamento matricial - visualização do campo potencial.....	51
Figura 58 - Comparação entre referenciais	52
Figura 59- Planeamento Matricial - Sequência de ações.....	54
Figura 60- Numeração das células vizinhas	55
Figura 61 - Planeamento Matricial, obtenção da direção [20]	55
Figura 62 - Planeamento Matricial - mudança de referencial (exemplo).....	56
Figura 63 - Planeamento Radial - Zona de interesse	57
Figura 64 - Planeamento Radial - Pontos de interesse	59
Figura 65 - Planeamento Radial - exemplo	60
Figura 66 - <i>Process Control</i> da Navegação por GPS	61

Figura 67 - Esquema do funcionamento global do planeamento por GPS.....	62
Figura 68 - Planeador Matricial - Circuito do ensaio do tipo 1	65
Figura 69 - Planeamento matricial - ensaio do tipo 1 – 1	65
Figura 70 - Planeamento matricial - ensaio do tipo 1 - 2.....	66
Figura 71 - Planeamento matricial - ensaio do tipo 1 – 3	66
Figura 72 - Planeador Matricial - Circuito do ensaio do tipo 2	67
Figura 73 - Planeamento matricial - ensaio do tipo 2 – 1	67
Figura 74 - Planeamento matricial - ensaio do tipo 2 – 2	68
Figura 75 - Planeamento matricial - ensaio do tipo 2 – 3	68
Figura 76 - Planeamento Matricial - Circuito do ensaio do tipo 3	69
Figura 77 - Planeamento matricial - ensaio do tipo 3 – 1	70
Figura 78 - Planeamento matricial - ensaio do tipo 3 – 2	70
Figura 79 - Planeamento matricial - ensaio do tipo 3 – 3	71
Figura 80 - Planeamento matricial - ensaio do tipo 3 – 4	71
Figura 81 - Planeamento Matricial - Circuito do ensaio do tipo 4	72
Figura 82 - Planeamento matricial - ensaio do tipo 4 –1.....	73
Figura 83 - Planeamento matricial - ensaio do tipo 4 – 2	73
Figura 84 - Planeamento matricial - ensaio do tipo 4 – 3	74
Figura 85 - Planeamento Matricial - Circuito do ensaio do tipo 5	75
Figura 86 - Planeamento Matricial - Ensaio do tipo 5-1	75
Figura 87 - Planeamento Matricial - Ensaio do tipo 5-2	76
Figura 88 - Planeamento radial - ensaio do tipo 1 – 1	77
Figura 89 - Planeamento radial - ensaio do tipo 1 – 2	78
Figura 90 - Planeamento radial - ensaio do tipo 2 – 1	79
Figura 91 - Planeamento radial - ensaio do tipo 2 – 2	79
Figura 92 - Planeamento radial - ensaio do tipo 3 –.1.....	80
Figura 93 - Planeamento radial - ensaio do tipo 3 – 2	81
Figura 94 - Planeamento radial - ensaio do tipo 3 – 3	81
Figura 95 - Planeamento Radial - Circuito do ensaio do tipo 3.....	82
Figura 96 - Planeamento radial - ensaio do tipo 4 – 1	83
Figura 97 - Planeamento radial - ensaio do tipo 4 – 2	83
Figura 98 - Planeamento radial - ensaio do tipo 4 – 3	84
Figura 99 - Planeamento radial - ensaio do tipo 5 - 1.a.....	85
Figura 100 - Planeamento radial - ensaio do tipo 5 – 1.b.....	85

Figura 101 - Planeamento radial - ensaio do tipo 5 – 2	86
Figura 102 - Planeamento radial - ensaio do tipo 5 – 3	86
Figura 103 - Planeamento radial - ensaio do tipo 5 – 4	87

Lista de Tabelas

Tabela 1 - Especificações de comunicação NMEA 0183	13
Tabela 2 – Exemplos de frases GPS.....	14
Tabela 3 – Características do recetor GPS [15]	28
Tabela 4 - Resistências utilizadas nos divisores resistivos	34
Tabela 5 - Planeamento Matricial – Orientações e distância das células destino	55
Tabela 6 - Planeamento Radial – Orientações das células destino.....	60
Tabela 7 - Monitorização das Baterias - ensaios realizados	64
Tabela 8 - Planeamento Matricial - Ensaios realizados do Tipo 1.....	65
Tabela 9 - Planeamento Matricial - Ensaios realizados do Tipo 2.....	67
Tabela 10 - Planeamento Matricial - Ensaios realizados do Tipo 3.....	69
Tabela 11 - Planeamento Matricial - Ensaios realizados do Tipo 4.....	72
Tabela 12 - Planeamento Matricial - Ensaios realizados do Tipo 5.....	75
Tabela 13 - Planeamento Radial – Ensaios realizados do tipo 1.....	77
Tabela 14 - Planeamento Radial - Ensaios realizados do tipo 2.....	78
Tabela 15 - Planeamento Radial - Ensaios realizados do tipo3.....	80
Tabela 16 - Planeamento Radial - Ensaios realizados do Tipo 4	82
Tabela 17 – Planeamento Radial – Ensaios realizados do tipo 5	85

Lista de Equações

(1) - Campos de Potencial – Potencial atrativo.....	21
(2) - Campos de Potencial – Potencial repulsivo.....	22
(3) - Campos de Potencial – Força atrativa	23
(4) - Campos de Potencial – Força repulsiva.....	23
(5) - Tensão de saída num divisor resistivo.....	33
(6) - Constante de conversão (Latitude)	46
(7) - Constante de conversão (Longitude)	46
(8) - Conversão Latitude-pixel,y.....	46
(9) - Conversão Longitude-pixel,x	46
(10) - Divisão de células no planeamento matricial.....	50
(11) - Distância em células dos obstáculos em x.....	53
(12) - Distância em células dos obstáculos em y	53
(13) - Colocação dos obstáculos no referencial global, x.....	53
(14) - Colocação dos obstáculos no referencial global, y.....	53
(15) - Planeamento matricial - Mudança de referencial.....	54
(16) - Raio do arco entre dois pontos	55
(17) - Ângulo de viragem para percorrer determinado arco	55
(18) - Fórmula Haversine, 1ª função	58
(19) - Fórmula Haversine, 2ª função	58
(20) - Fórmula Haversine, 3ª função	58
(21) - Cálculo da orientação entre dois pontos.....	58

Lista de Estruturas de Mensagens

Estrutura 1 - Exemplo da estrutura de uma mensagem	11
Estrutura 2 - Mensagem recebida pelo Base	37
Estrutura 3 - Mensagem publicada pelo Base (estado do robô)	37
Estrutura 4 - Mensagem dos módulos de aquisição de imagem	38
Estrutura 5 - Mensagens do laser	38
Estrutura 6 - Mensagem de comando para o reprodutor	40
Estrutura 7 - Mensagem do estado do reprodutor	40
Estrutura 8 - Mensagens do módulo de fusão	41
Estrutura 9 - Mensagem da monitorização das baterias	43
Estrutura 10 - Mensagem GPS-RMC	45
Estrutura 11 - Mensagem de orientação	47
Estrutura 12 - Mensagem de Navegação Inicial	48
Estrutura 13 - Mensagem do planeador	50



1. Introdução

O principal objetivo deste trabalho consiste na adaptação do robô da série ATLAS2000 para meios exteriores bem como na inclusão de um sistema de navegação por GPS. Este robô pertence ao grupo de estudo e investigação Atlas do Laboratório de Automação e Robótica (LAR) do Departamento de Engenharia Mecânica (DEM) da Universidade de Aveiro (UA), que se foca, entre outros, no estudo e desenvolvimento de sistemas de condução autónoma.

1.1. Apresentação do Projeto Atlas

Como referido, o robô em estudo pertence ao grupo Atlas que surgiu em 2002 [1] para competir em provas de condução autónoma, tendo sido, desde então, premiado várias vezes no Festival Nacional de Robótica com o primeiro e segundo prémio, obtidos com os robôs Atlas-2000 e Atlas-MV, representados respetivamente na Figura 1 e Figura 2. O grupo ainda detém o recorde de velocidade da competição, em que o Atlas-2009 conseguiu completar as duas voltas à pista num tempo de 42 segundos. Este festival decorre anualmente englobando várias vertentes da robótica, com competições, um encontro científico e demonstrações. Recentemente a equipa Atlas tem estado a usar o *know-how* adquirido ao longo dos anos com os carros de escala reduzida para desenvolver um automóvel de condução autónoma, o AtlasCar (Figura 3).

O primeiro robô da série Atlas 2000 foi fruto de um projeto de alunos do Departamento de Engenharia Mecânica no ano letivo 2002/2003, e tinham como objetivo competir em provas nacionais de condução autónoma. O projeto intitulou-se de Atlas, nome dado ao robô desenvolvido. Este robô possuía um chassis construído em madeira e alumínio, e o seu movimento era assegurado por duas rodas traseiras e a direção por uma roda dianteira. Este robô encontra-se ilustrado na Figura 4.



Figura 1 - Atlas 2000

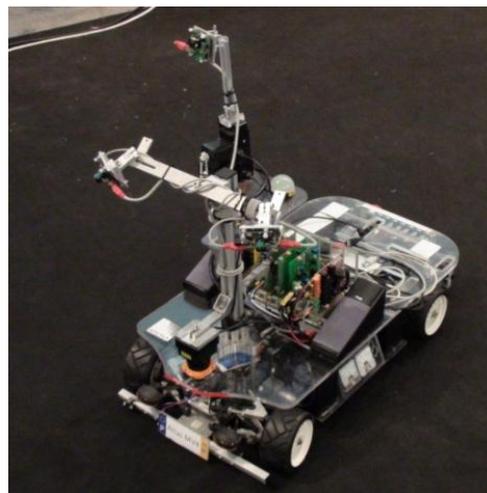


Figura 2 - Atlas MV



Figura 3 - Atlas Car1

Devido ao grande interesse do grupo, o projeto Atlas não cessou nesse ano, competindo desde então nos festivais anuais. No segundo ano do projeto, o robô sofreu ligeiras alterações e passou a ser denominado de Atlas II. Na evolução que se seguiu, o robô sofreu uma grande alteração física, tendo o grupo decidido usar um chassis de um carro de rádio modelismo de escala 1:4 para a base do robô, que se encontra ilustrado na Figura 5. Uma estrutura deste género oferece uma maior estabilidade e capacidade de movimentação ao robô, razão pela qual ainda se mantém no modelo atual. Todos os anos o robô sofre alterações físicas e a nível de *software*, sendo as últimas a inserção de um sistema de segurança com sensores ultrassons e uma reestruturação completa da eletrónica, trabalho desenvolvido na dissertação do aluno Bruno Andrade [2]. Na Figura 6 e Figura 7 pode-se averiguar a melhoria da organização e estruturação da eletrónica do robô.



Figura 4 - Estrutura do Atlas I/II [4]

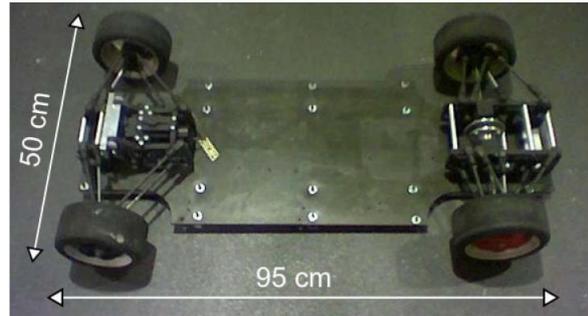


Figura 5 - Chassis do Atlas II

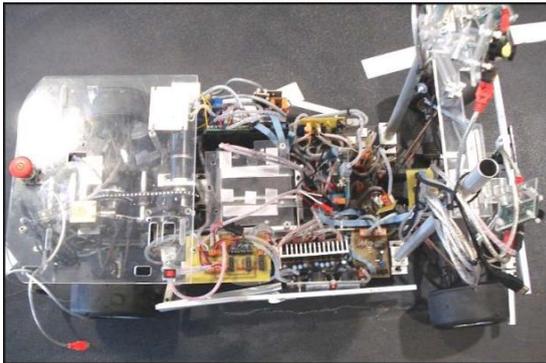


Figura 6 - Estado da eletrônica pré-2010 [2]

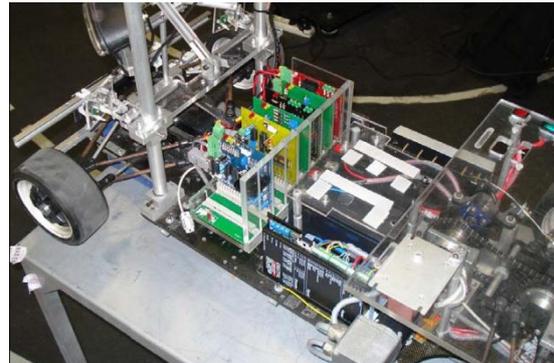


Figura 7 - Estado da eletrônica pós-2010 [2]

1.2. O Festival Nacional de Robótica

Como mencionado anteriormente, a equipa Atlas participava desde 2002 no Festival Nacional de Robótica nas competições de condução autónoma, a sua última participação foi na edição de 2011 ocorrido nas instalações do Instituto Superior Técnico em Lisboa, durante os dias de 7 a 10 de Abril de 2011 [3]. A competição de condução autónoma consiste em dar duas voltas completas a uma pista em forma de oito, como ilustrada na Figura 8, de modo a obter o melhor tempo possível. Desde 2011 a competição foi dividida em duas categorias, *rookie class* e *challenge class*, tendo o Atlas 2011 e o Atlas MV4 participado na última. A prova é dividida em três mangas que ocorrem em três dias consecutivos, sendo o grau de dificuldade aumentado de dia para dia. A primeira manga, a mais simples, consiste apenas em dar duas voltas completas à pista, sem ultrapassar os seus limites, e parar na meta final; não existem obstáculos nem é preciso respeitar qualquer sinal de trânsito. No segundo dia, a manga torna-se um pouco mais exigente, com obstáculos em lugares aleatórios na pista e os semáforos ativos, sendo necessário respeitá-los. Ao completar a segunda volta é requerido estacionar no parque de estacionamento que se encontra ao lado da pista. O terceiro dia, o mais exigente, contém dois grandes desafios, um em cada extremidade da pista. Um deles é uma zona de obras que obriga o veículo a sair da pista, guiando-se apenas por um caminho de cones de



sinalização. Esta tarefa é de grande dificuldade visto que obriga o robô a ultrapassar os limites da pista, que vai contra as suas regras de condução. Na extremidade oposta encontra-se um túnel, que põe à prova o desempenho dos veículos em ambientes sem iluminação. Estes desafios são ultrapassados com o auxílio de um sensor laser Hokuyo URG-04LX-UG01 *rangefinder* que se encontra na parte dianteira do robô, o laser possui um alcance de 5,6 metros a 240º [5].

No decorrer de cada manga as equipas têm direito a quatro tentativas à pista para conseguir um bom tempo, no final do terceiro dia são contabilizados os melhores tempos de cada manga. Em 2011 a equipa Atlas voltou a arrecadar prémios no festival; o Atlas-2011 conseguiu o primeiro lugar e o Atlas-MV4 o segundo na competição de condução autónoma. A equipa ainda participou no concurso FreeBots, competição em que são apresentados, diante de um júri, novas ideias e tecnologias na área da robótica. Assim, foi apresentando o AtlasCar1 e o sucesso desta apresentação conseguiu o primeiro lugar.

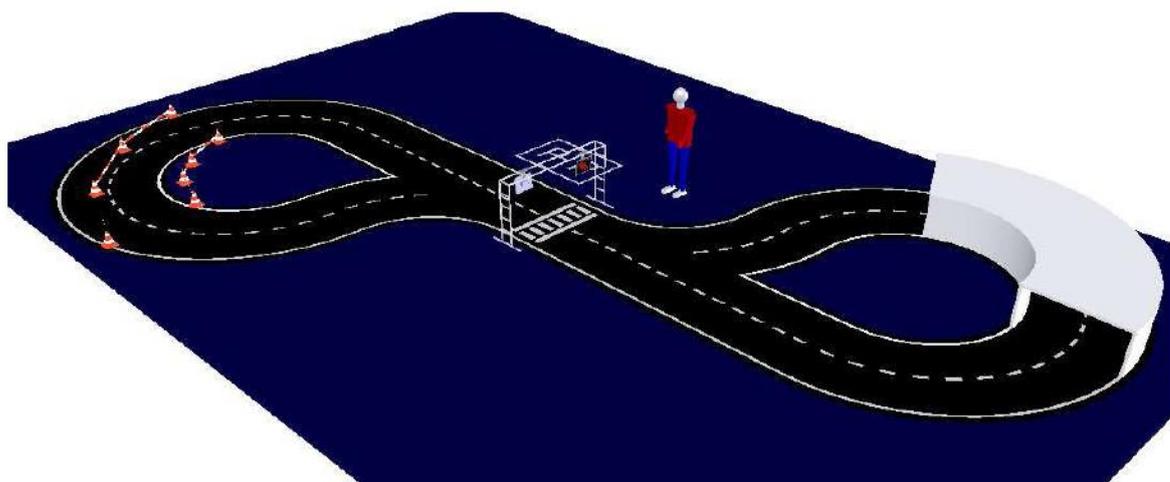


Figura 8 - Pista da competição de condução autónoma [2]

Durante a condução autónoma, o robô Atlas 2000 utiliza vários sistemas de deteção e reconhecimento de estrada e obstáculos, que têm vindo a evoluir progressivamente de ano para ano de modo a melhorar o desempenho global do robô. Esta evolução contante tem sido um fator determinante para o sucesso do grupo.

1.3. Objetivos/plano de intervenção

O objetivo principal desta dissertação resume-se à adaptação do robô da série ATLAS2000 para o ambiente exterior, incluindo a instalação de um sistema que permita uma navegação



autónoma por GPS. Para tal será instalado um sistema de receção GPS e desenvolvido *software* de localização e planeamento de trajetórias no campus da Universidade de Aveiro. A instalação deste recetor implica o desenvolvimento de *hardware* específico, que será detalhado no capítulo 3.2.

A intervenção mais notória será a mudança das rodas, que passarão de rodas convencionais de carros de modelismo para rodas pneumáticas de maior dimensão.

É também um objetivo a instalação de um sistema de verificação dos níveis de carga das baterias, que será útil para garantir que o veículo tenha sempre carga elétrica suficiente para voltar à base.

O plano de trabalho foi dividido em sete tarefas principais:

1 – Análise do problema e especificações: Familiarização com o sistema atual e suas limitações. Análise das possibilidades e requisitos para navegação em ambientes exteriores.

2 – Adaptações mecânicas: mudança do sistema das rodas para possibilitar a movimentação em pisos e ambientes mais irregulares e não estruturados.

3 – Migração, instalação e desenvolvimento de *software*: Migração e instalação de *software* para a arquitetura aberta CARMEN/IPC, incluindo a integração dos sistemas de perceção e navegação anteriormente desenvolvidos.

4 – Instalação de um sistema GPS: Instalação de um recetor de GPS e desenvolvimento de *software* para localização e de planeamento de caminhos.

5 – Instalação de um sistema de monitorização das baterias: Instalação de um sistema de verificação constante dos níveis das baterias capaz de comunicar com um computador.

6 – Ensaios em ambiente exterior: Ensaios no campus da UA e noutros locais.

7 – Escrita de documentação e da dissertação.

1.4. Estrutura da dissertação

Esta dissertação encontra-se dividida em seis capítulos, sendo o primeiro de introdução, seguido pelos capítulos que relatam as adaptações mecânicas e as alterações a nível de *software* que o robô



sofreu. De seguida serão descritos os ensaios realizados e os seus respetivos resultados. Por fim serão apresentadas as principais conclusões extraídas e também propostas para trabalhos futuros para dar seguimento ao trabalho desenvolvido. A dissertação finaliza com dois capítulos de apoio, sendo estas as de referências utilizadas e um de anexos.



2. Arquiteturas de *Software* e Navegação

Neste capítulo serão abordados e explicados os conceitos e sistemas que serão implementados no Atlas -2011. Até à data o robô utilizava um programa único que corria ciclicamente para adquirir dados, processá-los e acionar os atuadores, tornando-o num programa muito extenso, de difícil edição e exposto a potenciais falhas. Decidiu-se então adotar uma arquitetura modular, separando o programa em módulos e implementar a filosofia CARMEN [6], onde cada módulo é encarregue de executar uma tarefa específica e cujos resultados podem ser encaminhados para os restantes módulos, que se encontram a operar em simultâneo, através da troca de mensagens. A arquitetura modular permite a fácil edição do código de um módulo; a filosofia CARMEN será descrita mais detalhadamente no capítulo 2.1. Será também feita uma ligeira introdução ao conceito de GPS, explicando o seu funcionamento, aplicações possíveis e as suas vantagens. Por fim, serão descritas algumas filosofias de planeamento de navegação para robôs autónomos.

2.1. CARMEN/IPC

CARMEN (*Carnegie Mellon Robot Navigation Toolkit*) [6] é uma coleção de *software open-source* desenvolvida para o controlo de robôs móveis. É um sistema modular destinado a oferecer ordens básicas de navegação: controlo sensorial e de base, registo de dados, desvio de obstáculos, localização, planeamento de trajetórias e mapeamento. A comunicação entre programas CARMEN é tratada por um pacote separado denominado de IPC (*Inter-process Communication*) [7] que é realizada através de ligações TCP/IP que, com muita facilidade, consegue transmitir dados entre dois processos que estejam a correr em simultâneo. Este intercâmbio de informação baseia-se pelas regras de um sistema de publicadores e subscritores [8].

A informação trocada entre módulos tem de ser classificada para que um módulo receba apenas o que requer e mais nenhuma informação adicional. Isto é conseguido agrupando a informação em mensagens específicas; assim, se um módulo estiver “interessado” em determinada informação deve pedir para receber a mensagem correspondente; quando ficar subscrito àquela mensagem, recebe-a sempre que for publicada. Esta troca de informação pode ser efetuada de diversas formas, que se encontram descritas de seguida.



Método da publicação/subscrição

Este método de troca de mensagens é o mais simples dos abordados; o seu funcionamento geral encontra-se ilustrado na Figura 9. Nesta situação, o módulo publicador gera informação que é compactada numa mensagem de certo tipo, “A” por exemplo. Todos os outros módulos interessados nalguma informação compactada na mensagem do tipo “A” têm de subscrevê-la. Quando esta informação se encontra disponível, o módulo publicador divulga a mensagem do tipo “A” que será redirecionado por IPC para todos os módulos que previamente a subscreveram. Uma desvantagem é que o módulo publicador requer que os módulos subscritos recebam todas as mensagens do tipo “A”, mesmo se não as quiser todas. Isto poderá tornar-se crítico caso o tempo de ciclo do publicador seja inferior ao do subscritor.

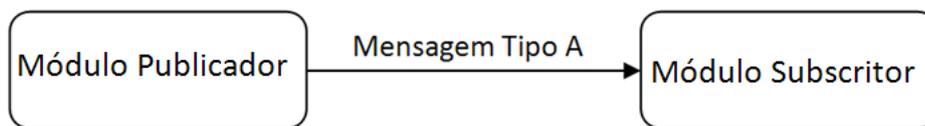


Figura 9 - Troca de mensagem básica em IPC

A Figura 10 demonstra uma situação em que o módulo subscritor, à direita, não consegue processar os dados recebidos antes da receção de novas mensagens, isto fará com que haja uma fila crescente de mensagens que poderá levar a uma sobrecarga do módulo central do IPC e por ventura uma eventual falha do sistema, especialmente em mensagens grandes (centenas de kilobytes). Assim, a metodologia de publicação/subscrição de mensagens apenas deve ser implementada caso o subscritor seja mais rápido que o publicador, ou se a rotina de tratamento da mensagem recebida for muito rápida.

Método do pedido/resposta

Como referido, o método publicação/subscrição é apropriado para o intercâmbio de mensagens pequenas ou para módulos com processamentos rápidos, que não é o caso dos robôs ATLAS, que realizam vários processamentos e trocas de imagens. Assim implementou-se uma segunda metodologia disponível no IPC, onde o módulo servidor (*server module*) apenas publica a mensagem quando pedido por outro módulo, denominado de módulo requerente (*query module*). Assim, o ritmo de intercâmbio de mensagens é estabelecido pelo módulo requerente em vez de ser definida arbitrariamente pelo servidor.

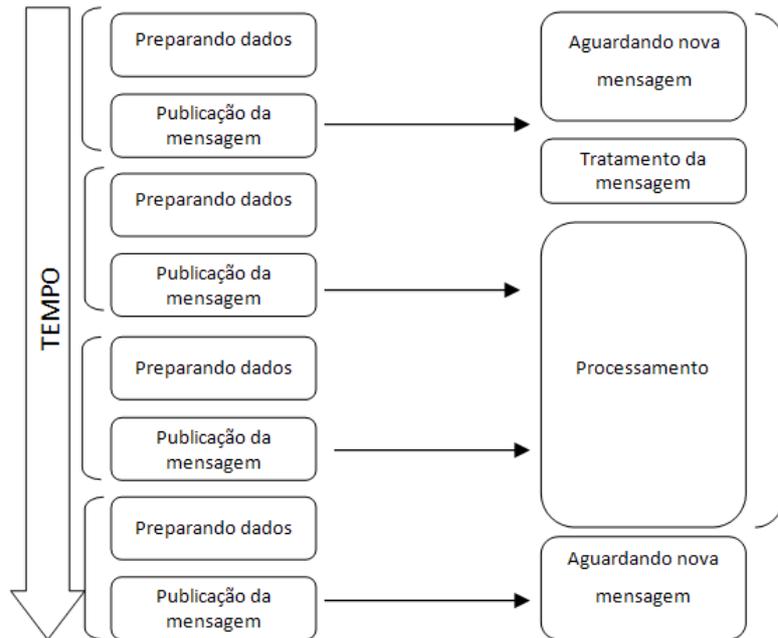


Figura 10 - Exemplo de sobrecarga numa comunicação IPC

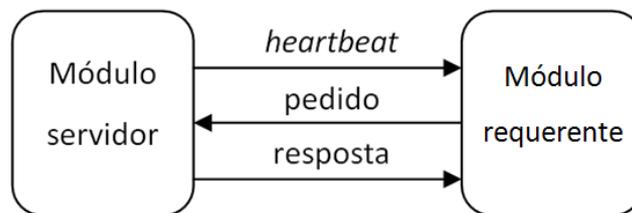


Figura 11 - IPC - Comunicação com pedido/resposta

Para suportar este método de intercâmbio três mensagens são definidas: o batimento (*heartbeat*), o pedido (*query*) e a resposta (*response*), o intercâmbio destas encontra-se ilustrado na Figura 11. A mensagem de *heartbeat* indica que o servidor possui uma nova informação e está apto para o enviar, é uma mensagem pequena como uma mera notificação para quem está interessado na nova informação. Estas mensagens são emitidas pelo módulo servidor utilizando a técnica publicação/subscrição, permitindo assim que vários módulos sejam informados em simultâneo. A mensagem de pedido, também de pequena dimensão, é enviada pelo módulo requerente ao servidor, e após este pedido o módulo aguarda a resposta proveniente do servidor. Tanto o pedido como a resposta são enviadas por uma comunicação *peer to peer*. Mesmo sendo mais complexo, este sistema é vantajoso na transmissão de mensagens grandes, visto que esta apenas é efetuada quando o módulo precisa da informação, reduzindo assim o tráfego de mensagens. Se o módulo requerente for mais rápido do que o módulo servidor, as mensagens *heartbeat* asseguram que nenhum pedido seja feito a não ser que uma nova informação esteja disponível do lado do servidor.



A cadeia de mensagens encontra-se ilustrada na Figura 12, em que na coluna do lado esquerdo encontra-se o módulo servidor e na do lado direito o módulo requerente. As mensagens de *heartbeat* e de pedido são mensagens que sincronizam ambos os módulos de modo a que as mensagens sejam apenas trocadas quando pedidas. Neste caso, a complexidade é compensada pelo não envio de mensagens grandes não desejadas. Assim, este sistema apenas deve ser implementado quando as mensagens são de grande dimensão.

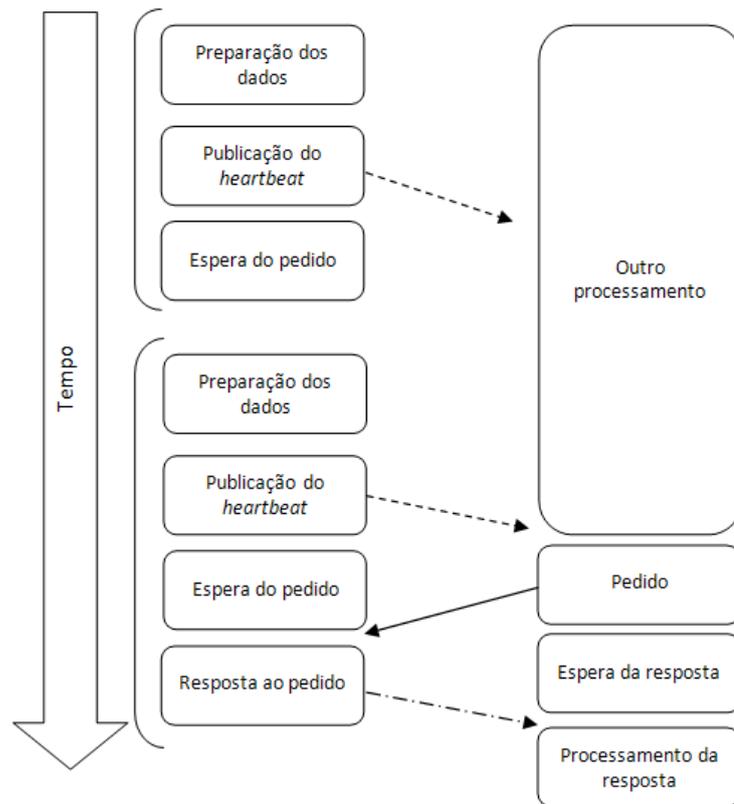


Figura 12 – Técnica de pedido/resposta com *heartbeat*

Método pedido/resposta com memória partilhada

A desvantagem do método pedido/resposta consiste na dedicação do módulo servidor responder a um pedido de um módulo específico, limitando a comunicação por esta ser *peer-to-peer*. O envio, pelo módulo servidor, de uma grande quantidade de informação para vários módulos, aumenta o tráfego de mensagens, proporcionalmente com o número de módulos requerentes. Se este número for suficientemente grande, o módulo servidor poderá não ser capaz responder a todos no tempo definido. Este problema foi resolvido com o desenvolvimento de um novo método, onde o módulo servidor escreve a mensagem contendo a resposta num endereço de memória partilhada.



Esta técnica usufrui das funcionalidades de compactação/descompactação (*marshal/unmarshal*) do IPC. *Marshaling* é o processo de transformar a mensagem numa *array* linear de *bytes* configurável e de fácil reversibilidade, as mensagens são definidas como estruturas de linguagem C, mas antes de serem enviadas têm de ser transformadas, compactadas, em *arrays* de *bytes*. Para estas serem descompactadas nas mensagens corretas, o formato da sua estrutura é previamente definida, na Estrutura 1 encontra-se um exemplo de estrutura de uma mensagem.

```
Typedef struct
{
    int var1;
    char var2;
    int var3[2];
} type_msgA

#define msgA_format "{int,char,int[2]}"
```

Estrutura 1 - Exemplo da estrutura de uma mensagem

No caso da Estrutura 1, após a compactação, a mensagem tem o formato da estrutura *msgA_format*, sendo formada por um inteiro seguido por um caractere seguido por um *array* de 2 inteiros. Os módulos utilizam o IPC para compactar todas as estruturas das mensagens que enviam e descompactar as que recebem. Nesta troca particular de informação, o módulo servidor aloca uma memória partilhada com o tamanho da mensagem, guardando esta mensagem diretamente nesta memória partilhada. Após esta operação, uma mensagem de *heartbeat* é publicada indicando que uma nova informação se encontra disponível. O módulo de pedido agrupa-se ao segmento de memória partilhada e descompacta a informação numa variável do tipo *type_msgA*, a operação de agrupamento requer o endereço da memória partilhada. Todo este processo encontra-se esquematizado na Figura 13, onde no lado esquerdo se encontra o módulo servidor que se encarrega de informar o módulo requerente do endereço da memória partilhada, onde será armazenada a informação, de compactar a mensagem e de publicar o *heartbeat* a informar a disponibilidade de informação nova. Do lado direito encontra-se o módulo requerente que solicita ao servidor o endereço da memória partilhada onde será guardada a informação desejada. A este propósito uma nova mensagem é definida contendo a informação da memória partilhada, o seu endereço e dimensão. Esta mensagem é indicada pelo servidor durante o procedimento de inicialização do módulo de pedido. Posteriormente, em qualquer altura que um *heartbeat* é recebido, o módulo de pedido apenas descompacta a informação da memória partilhada e obtém uma cópia da mensagem. A limitação deste método reside no facto de não ser baseado em TCP/IP, não funcionando quando os processos se encontram a correr em máquinas separadas. Não



obstante, este método é muito rápido e resolve os problemas que possam surgir ao usar vários módulos de pedido (*query modules*).

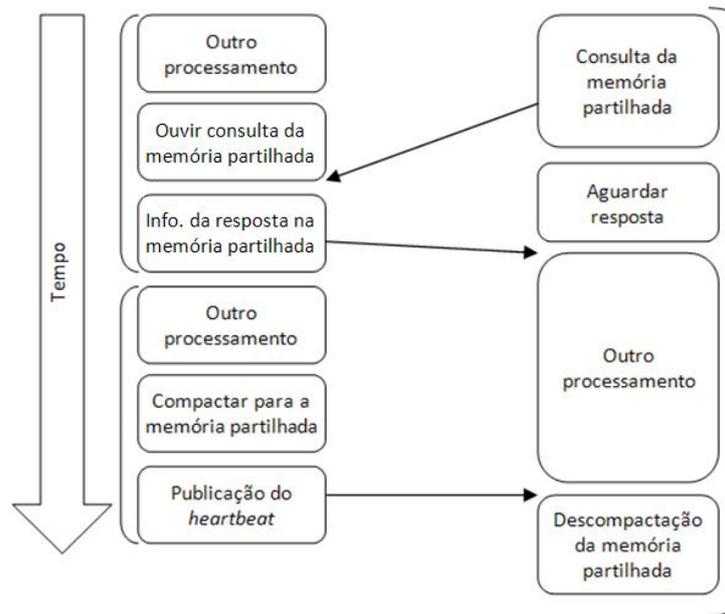


Figura 13 – Método de partilha com memória partilhada

2.2. GPS

Atualmente o conceito de GPS encontra-se tão divulgado que podemos encontrar sistemas que o utilizam na maioria dos telemóveis e automóveis do quotidiano. O sistema de posicionamento global (*Global Positioning System*) nasceu nos anos 60 num projeto que envolveu a NASA e o Departamento de Transportes do Estados Unidos da América; este projeto tinha como principal objetivo criar um sistema de posicionamento, através de satélites, para apoiar os serviços militares. Presentemente, o GPS ainda funciona com os mesmos critérios estabelecidos nos anos 60. O sistema oferece, com precisão e em qualquer parte do globo, uma posição a três dimensões e a velocidade a um utilizador que tenha o recetor apropriado. Este posicionamento é realizado com o auxílio de uma constelação de 24 satélites organizados em 6 órbitas com 4 satélites por órbita, ilustrados na Figura 14, utilizando um tempo universal sincronizado entre eles (UTC – *Coordinated Universal Time*) [9]. A posição do utilizador é calculada através da distância que este se encontra de cada um dos satélites conectados e, através de trigonometria, a posição do utilizador é determinada no referencial global. De modo a obter uma leitura correta é necessário estar em comunicação com pelo menos 4 satélites, teoricamente 3 seriam suficientes mas visto que são usados valores elevados nos cálculos, o mais pequeno erro de cálculo resultará num erro considerável de localização no espaço. Assim,



recetores utilizam 4 ou mais satélites para calcular uma localização mais correta. Além do sistema GPS criado pelos americanos existem outros sistemas que se encontram em fase de desenvolvimento; o *Glonass* é um sistema russo que tem um funcionamento semelhante ao americano, utilizando também 24 satélites. Ainda existem os sistemas da *Galileo* e *Compass*, da União Europeia e da China, respetivamente.

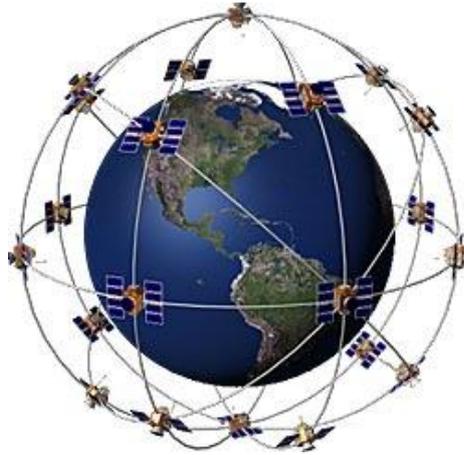


Figura 14 - GPS - Constelação de satélites

A *National Marine Electronics Association* (NMEA) [9] desenvolveu um protocolo para facilitar a comunicação entre recetores de GPS e qualquer outro instrumento eletrónico, o protocolo NMEA 0183. Que inclui um conjunto de especificações que definem o tipo de comunicação e o formato em que a informação é transmitida; na Tabela 1 encontram-se as principais características da comunicação deste protocolo. A informação proveniente dos satélites é separada em frases consoante o tipo de dados, onde cada uma destas frases é iniciada com o símbolo de cifrão, “\$”, e os dados são separados por vírgulas; na Tabela 2 podemos encontrar exemplos destas frases.

Tabela 1 - Especificações de comunicação NMEA 0183

Velocidade de transmissão	4800
Nº de bits	8
Paridade	Nulo
Stop Bits	1
Handshake	Nulo



Tabela 2 – Exemplos de frases GPS

Sigla	Nome	Exemplo de frase
GGA	Global Positioning System Fixed Data	\$GPGGA,161229.487,3723.2475,N,12158.3416,W,1,07,1.0,9.0,M,,,,,0000*18
GSA	DOP and Active Satellites	\$GPGSA,A,3,07,02,26,27,09,04,15,,,,,1.8,1.0,1.5*33
GSV	Satellites in View	\$GPGSV,2,1,07,07,79,048,42,02,51,062,43,26,36,256,42,27,27,138,42*71
RMC	Recommended Minimum Specific GNSS Data	\$GPRMC,161229.487,A,3723.2475,N,12158.3416,W,0.13,309.62,120598,,*10
VTG	Course Over Ground and Ground Speed	\$GPVTG,309.62,T,,M,0.13,N,0.2,K*6E
GLL	Geographic Position- Latitude/Longitude	\$GPGLL,3723.2475,N,12158.3416,W,161229.487,A*2C

2.3. Planeamento de caminhos

Este projeto inclui a utilização de um método de planeamento de caminhos; assim, realizou-se uma pesquisa de métodos e técnicas existentes para resolver problemas de planeamento. Descobriu-se um vasto número de métodos para resolver problemas básicos de planeamento de caminhos de robôs móveis. Muitos não resolvem situações genéricas, sendo necessário que o espaço de trabalho seja de duas dimensões e que os objetos sejam considerados polígonos. Ao longo deste trabalho, apenas algumas destas técnicas serão mencionadas e explicadas. Em geral assentam no mesmo princípio de planeamento de caminhos, que se divide em pré-processamento e processamento de busca. O primeiro consiste em obter informação do espaço vazio em torno do robô e elaborar uma conectividade viável, podendo ser em grafo, árvore, matriz, ou qualquer outro elemento que permita dividir e classificar regiões. Esta informação pode ser previamente conhecida ou obtida, dinamicamente, a partir de sensores instalados no robô. O segundo princípio consiste na análise destas regiões e optar pelo caminho que melhor satisfaça as necessidades do robô. Serão referidas três famílias de métodos [10] [11] [12] [13]:

- Decomposição em células
- Mapa de estradas
- Campos de potencial



2.3.1. Método de Decomposição em células

O método de decomposição em células é o método de planeamento de movimentação mais estudado até à data. Consiste na divisão do espaço livre à volta do robô em regiões simples, chamadas de células, de forma que um caminho entre quaisquer dois pontos na mesma célula possa facilmente ser obtido. Como já referido, os obstáculos são sempre considerados corpos bidimensionais poligonais. Em seguida é efetuado um grafo representante das células adjacentes, representando os caminhos possíveis que o robô pode optar. Na fase de pré-processamento é realizada a repartição do espaço vazio e a construção do grafo, denominado grafo de conectividade. Seguidamente, na fase de processamento, é efetuada uma pesquisa de células adjacentes entre a célula inicial e a célula final. Esta sequência de células é denominada de canal, do qual pode ser calculado um caminho contínuo desde do ponto de partida até ao destino. Os métodos baseados em decomposição em células podem ser divididos ainda em exatos e aproximados:

- Métodos exatos de decomposição em células decompõem o espaço livre num conjunto de células cuja união cobre exatamente todo o espaço livre.
- Métodos aproximados de decomposição em células dividem o espaço livre num conjunto de células de forma predefinida cuja união pode não cobrir inteiramente o espaço livre.

Os métodos exatos são ditos completos, isto é, permitem que um caminho entre duas quaisquer posições seja obtido sempre que for possível, dado que seja utilizado um algoritmo de busca apropriado. Na Figura 15 encontra-se uma explicação ilustrativa desta técnica, sendo o espaço vazio denominado de “S”, a posição inicial do robô representado pelo círculo vermelho e o ponto destino pelo azul. Esta decomposição de células pode ser efetuada através de diversas técnicas, sendo a única diferença entre estas a forma de dividir o espaço, e sendo as mais usadas as de decomposição em células poligonais e em trapezoides verticais.

Os métodos aproximados podem não ser completos, pois podem, dependendo da precisão utilizada, não encontrar um caminho entre duas posições mesmo que este exista. Contudo, os métodos aproximados são mais simples e os mais utilizados na prática. A principal diferença entre o método exato e o aproximado reside na forma da célula, sendo previamente definida no método aproximado. Consequentemente, na maioria das vezes não é possível modelar o ambiente de forma exata como ele é, sendo necessário que algumas aproximações sejam feitas. A Figura 16 ilustra três situações do método aproximado de decomposição em células, onde na imagem (a) é utilizada uma



célula que permite obter o percurso entre a partida, q_{start} , e o destino, q_{end} . Por sua vez, na situação (b1) é utilizada uma célula maior que impossibilita a realização de um percurso contínuo desde a partida até ao destino, sendo necessário utilizar uma célula mais adequada, como ilustrado na imagem (b2).

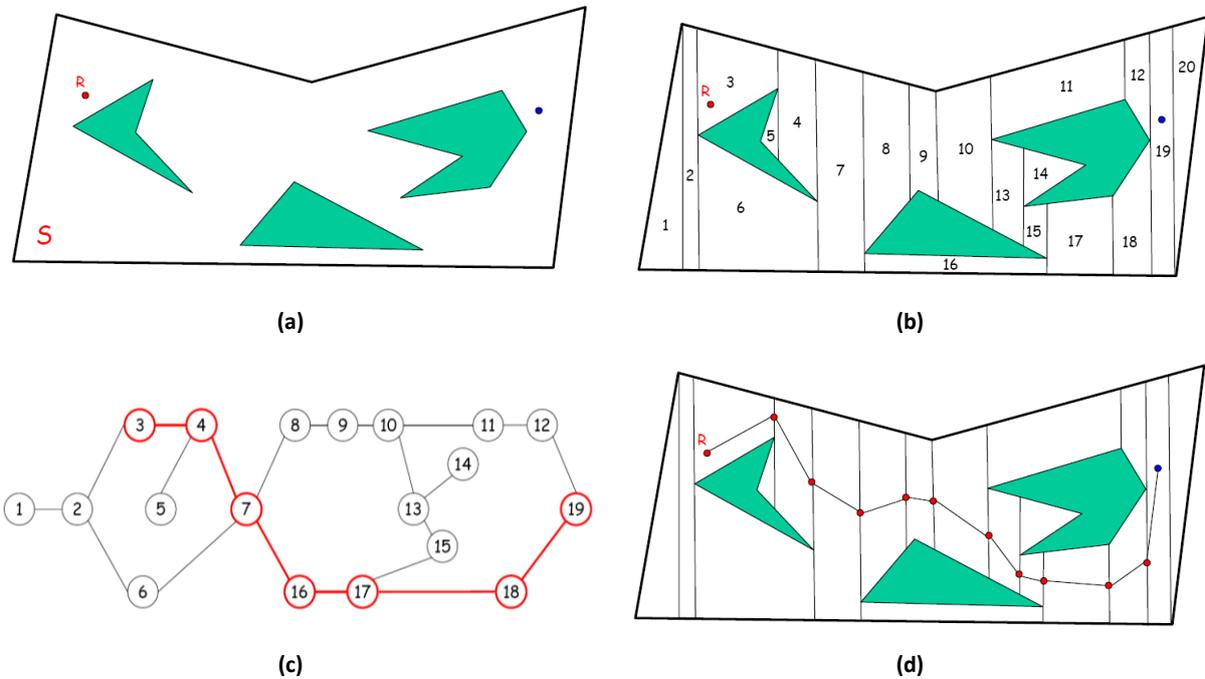


Figura 15 - Método de decomposição em células (exato)

(a) - Espaço "S"

(b) - Divisão do espaço "S" em células

(c) - Grafo de conectividade com percurso escolhido

(d) - Percurso final obtido

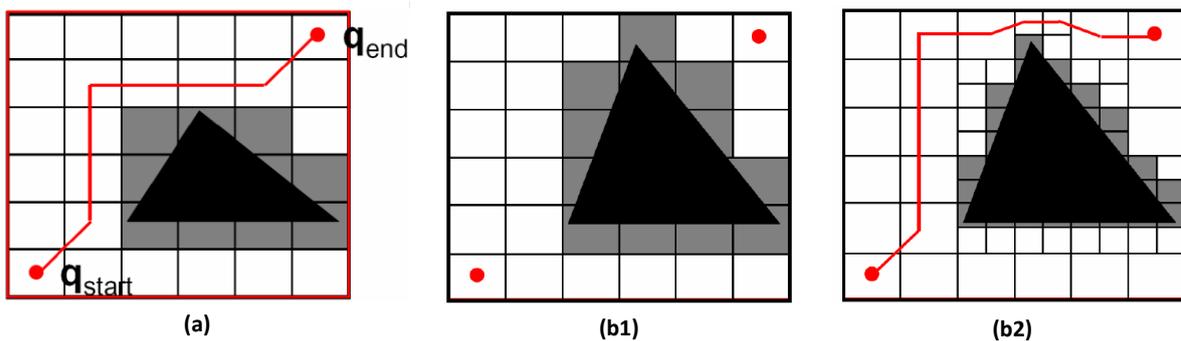


Figura 16 - Decomposição em células (aproximado)

Vale ainda referir que com o método aproximado pode-se ajustar à precisão desejada mudando apenas o formato e tamanho das células, tornando-o uma técnica bastante flexível. Entretanto, é importante salientar que a possibilidade de ajuste da precisão deste método está intimamente relacionada à quantidade de espaço e de tempo de execução do planeador de



caminhos. Além das diferenças mencionadas, o método aproximado é semelhante ao exato. Ou seja, após ser feita a decomposição do ambiente em células, constrói-se o grafo de conectividade e descobre-se neste uma sequência de células, na qual células subsequentes sejam adjacentes, desde à primeira até a última.

2.3.2. Método de mapa de estradas (*roadmap*)

Este planeamento consiste em capturar a conectividade do espaço livre, em torno do robô, numa rede unidimensional de curvas, chamada *roadmap*, esta é a fase de pré-processamento. A rede de curvas obtida representa a conectividade entre os espaços livres do meio em análise. Após a sua construção esta é definida como um conjunto de percursos estandardizados, reduzindo o planeamento de trajetórias à ligação entre os pontos de interesse. Se construída, a trajetória é definida por três subcaminhos: o subcaminho a ligar o ponto inicial ao mapa de estradas, o subcaminho do próprio mapa de estradas e por fim, o subcaminho que liga o mapa de estradas ao ponto final. Existem algumas técnicas de processamento de busca para este método:

- Grafo de visibilidade
- Diagrama de Voronoi

➤ Grafo de visibilidade

Este foi um dos primeiros métodos de planeamento de caminhos a ser desenvolvido e aplica-se principalmente a espaços bidimensionais com obstáculos de regiões poligonais. Compõe-se por todas as sequências de segmentos de reta que unem os vértices de obstáculos e os pontos de partida e chegada, sem atravessar nenhum destes. A escolha de caminho é feita pelo percurso mais curto, ou seja, com menos segmentos. Depois de escolhido o percurso final, esta técnica requer processamento adicional para afastar as trajetórias das esquinas dos obstáculos. Usualmente é realizada uma ampliação dos obstáculos antes da realização do grafo de modo a evitar este pós processamento. O referido método tem a vantagem de proporcionar o caminho mais curto entre a partida e o destino.

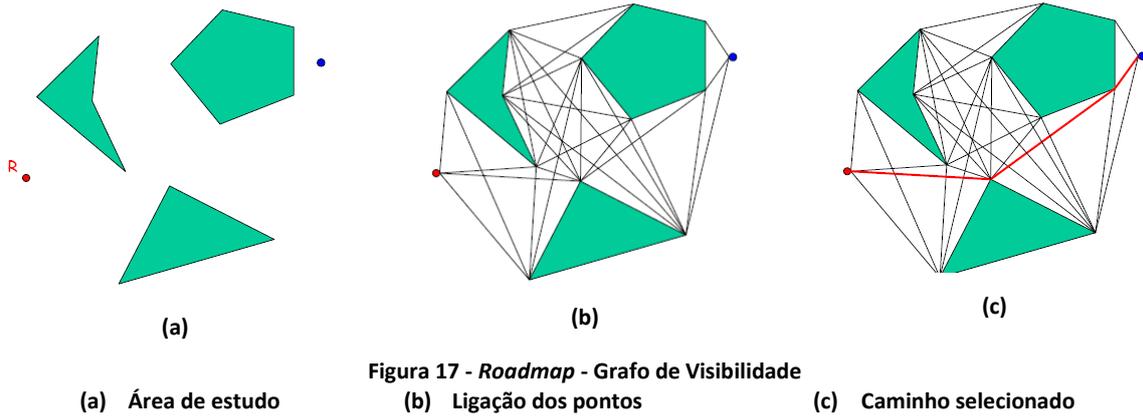


Figura 17 - Roadmap - Grafo de Visibilidade

➤ Diagrama de Voronoi

O diagrama de Voronoi representa a interseção dos meios-espaços dos todos pontos que delimitam os obstáculos e o meio, originando numa divisão em células, denominados de células de Voronoi. Assim, os segmentos do diagrama de Voronoi representam todos os pontos, num plano, que estejam equidistantes a dois pontos delimitadoras dos obstáculos ou do meio. O diagrama obtido representa a rede de curvas, o *roadmap*, e raramente este se sobrepõe com os pontos de origem ou destino do problema, sendo necessário criar dois pequenos subcaminhos para completar o diagrama, como podemos observar na Figura 18. Contudo, este método apresenta desvantagens, sendo as mais relevantes a dificuldade de o computar em problemas de maiores dimensões ou em que os obstáculos não são considerados polígonos. Ainda pode-se tonar num sistema instável, dado que uma pequena mudança na configuração dos obstáculos pode levar a uma grande mudança no diagrama obtido.

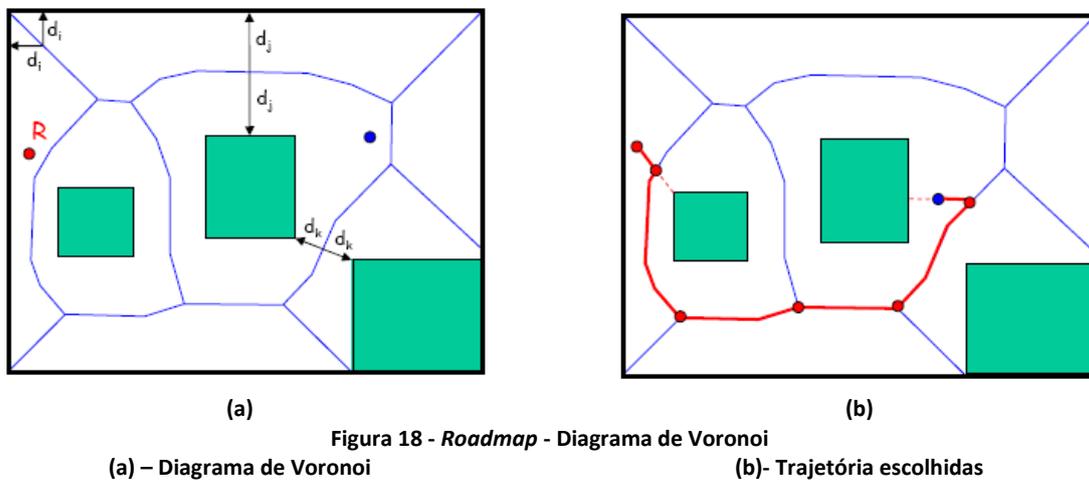
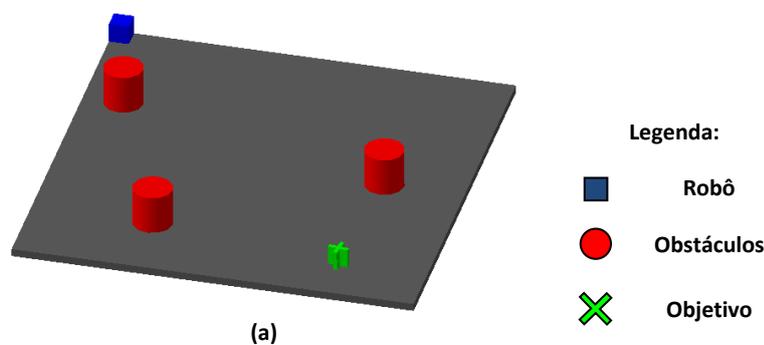


Figura 18 - Roadmap - Diagrama de Voronoi



2.3.3. Método dos Campos de Potencial

O método dos campos de potencial consiste numa função definida sobre a região livre do espaço, onde o robô é repelido pelos obstáculos e atraído pelo objetivo. A fase de pré processamento desta técnica baseia-se na definição da função do campo de potencial sobre todo o espaço S , normalmente conseguida recorrendo à discretização do espaço numa grelha de valores. A segunda fase, a de processamento de pesquisa, consiste na pesquisa de um caminho sobre a grelha usando o campo de potencial. O algoritmo de pesquisa é normalmente heurístico, estando sujeito a mínimos locais. O campo de potencial consiste na soma de duas “energias”, o potencial atrator e o potencial repulsor. Sobre esta soma define-se um gradiente de navegação que dirige o movimento local do robô em direção ao objetivo, associando cada célula da grelha ao respetivo valor de campo de potencial. A metáfora que a terminologia sugere é que o robô, representado por um ponto, seja uma partícula movendo-se sob a influência de um campo potencial artificial gerado pelo ponto objetivo e pelos obstáculos. O ponto objetivo gera um campo potencial que atrai o robô e por sua vez, os obstáculos geram um que provoca a repulsão do robô. O conceito resume-se a que em cada célula, a direção do movimento do robô seja determinada pela força resultante do campo potencial total nesse determinado instante. Este método tem a vantagem de ser de fácil visualização e implementação. Considerando o espaço em estudo como um plano, em que os obstáculos formam montanhas e o destino um vale, o robô, considerado um esfera, terá tendência a se deslocar para o destino, desviando-se dos obstáculos, por efeito da gravidade. A Figura 19 ilustra a noção de potencial repulsor e atrator, bem como a combinação deles num problema genérico. O potencial atrator (b) é um paraboloide com o ponto mínimo localizado no objetivo, já o potencial repulsor (c) é diferente de zero somente a partir de uma determinada distância dos obstáculos, aumentando com a proximidade destes. O caminho entre a origem e o destino é construído pela direção oposta à do gradiente do potencial resultante em cada ponto.



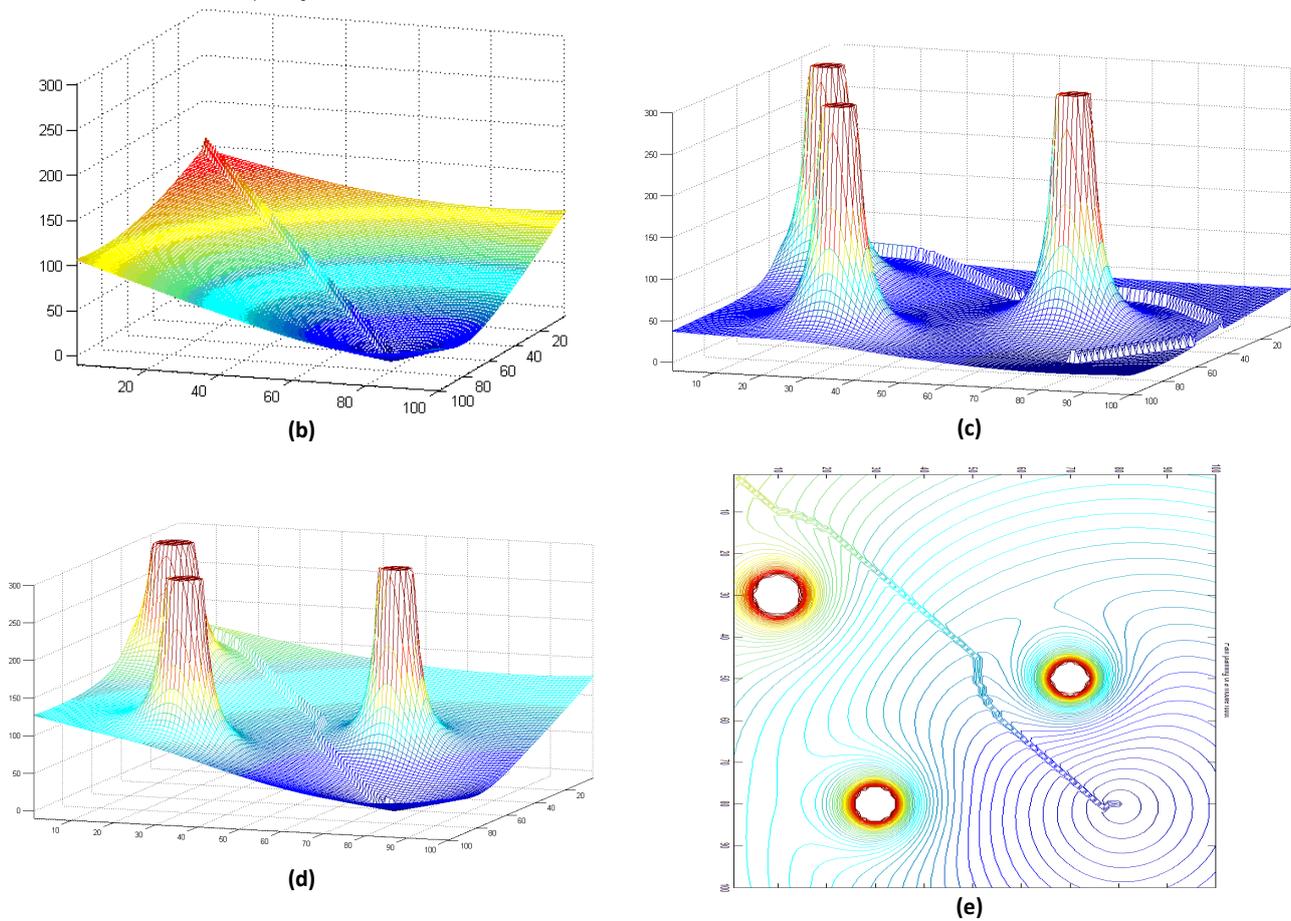


Figura 19 - Campos de Potencial

(a) – Situação em análise

(b) – Campo atrator

(c) – Campo repulsor

(d) – Junção dos campos

(e) – Vista superior

Em comparação com outros métodos, o de campo potencial pode ser bastante eficiente, porém, possui uma grande desvantagem. Como é baseado em métodos rápidos de otimização, pode levar o robô a um ponto que seja um mínimo local da função potencial, colocando-o numa situação indesejada. Um ponto de mínimo local é um ponto em que o potencial é mínimo, mas não é o mínimo global (o ponto objetivo). A Figura 20 ilustra esta situação, onde o robô é atraído pelo destino e repelido pelo obstáculo que se encontra entre eles, gerando assim um mínimo local que impossibilita a chegada do robô ao ponto destino, o mínimo global.

Uma forma de abordar este problema é pela construção de funções de potencial que não possuam mínimos locais, pelo menos no subconjunto conectado do espaço livre onde se encontra o ponto objetivo. Contudo, isto nunca é uma tarefa fácil, principalmente em espaços de configurações com obstáculos dinâmicos. Uma outra técnica empregada é adicionar, ao esquema básico de campo potencial, um mecanismo suficientemente poderoso para escapar de mínimos locais. Há vários métodos de otimização para este tipo de problemas, em especial as meta-heurísticas, como



simulated annealing, algoritmos genéticos e GRASP (*Greedy Randomized Adaptive Search Procedures*) [14]. Entretanto, não há nenhum método que garanta, para qualquer função de otimização, que não se cairá em um mínimo local.

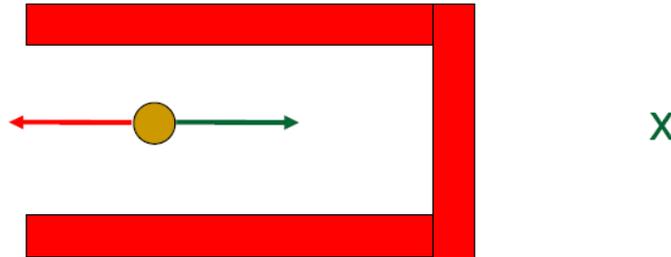


Figura 20 - Campo de Potencial - Mínimo local

➤ Definição analítica

O método dos Campos de Potencial possui a vantagem de usar fórmulas de cálculo flexíveis, que podem ser adaptadas caso a caso, apenas têm de garantir as seguintes condições:

- O potencial atrator diminui com a proximidade ao destino
- O potencial repulsor aumenta com a proximidade dos obstáculos

➤ Potenciais

O campo de potencial é o somatório dos dois potenciais, o atrator e o repulsor, que avaliam o nível de atração de um ponto pelo ponto destino e a repulsão que este sofre pelos obstáculos próximos.

Potencial atrator

$$U_{att}(q) = \frac{1}{2} \xi \rho_{goal}^2(q) = \frac{1}{2} \xi \|q - q_{goal}\|^2 \quad (1)$$

A equação (1) permite o cálculo do potencial atrator, onde U_{att} corresponde ao potencial atrator de um ponto q em relação ao ponto objetivo, q_{goal} , ξ à constante atratora e $\rho_{goal}(q)$ à distância



entre o ponto q e o ponto objetivo. A constante atratora define a força de atração do potencial, está diretamente relacionada com o valor de potencial atrator calculado;

Potencial repulsor

$$U_{rep,i}(q) = \begin{cases} \frac{1}{2}\eta \left(\frac{1}{\rho(q, q_i)} - \frac{1}{\rho_0} \right)^2 & \Leftarrow \rho(q, q_i) < \rho_0 \\ 0 & \Leftarrow \rho(q, q_i) > \rho_0 \end{cases} \quad (2)$$

O potencial repulsor apenas possui valores diferentes de zero nas proximidades dos obstáculos, e está relacionado com a distância entre o ponto em estudo e os obstáculos. Se a distância entre um ponto q e um obstáculo q_i for superior à distância de influência previamente definida, ρ_0 , o potencial repulsor deste obstáculo neste ponto será nulo. Esta condição está definida na equação (2), em que $U_{rep,i}(q)$ corresponde ao potencial repulsor do obstáculo i em relação ao ponto q . Tal como no potencial atrator, este potencial também possui uma constante: a constante repulsora η que define o peso da repulsão que os obstáculos irão possuir no campo de potencial. Por norma, esta constante é definida com valores superiores à constante de atração, de modo a elevar o potencial repulsor dos obstáculos ao potencial atrator do objetivo. A distância entre o ponto em estudo e o obstáculo i é representado por $\rho(q, q_i)$ e a distância de influência por ρ_0 . Esta é predefinida e define a distância mínima necessária para um obstáculo afetar o potencial repulsor de um ponto. O potencial repulsor aumenta quadraticamente com a proximidade do ponto ao obstáculo.

➤ Forças

A força do campo potencial de um dado ponto é dada pelo seu gradiente, ou seja, a alteração no valor da uma quantidade por unidade de espaço. Neste caso mais específico, representa a alteração do valor do campo de potencial de célula para célula. A escolha da trajetória é realizada pelo gradiente negativo mais elevado, significando que a próxima célula possui o potencial mais baixo da vizinhança analisada. Além de auxiliar na escolha de caminho, também pode ser útil na definição da velocidade, quanto maior a diferença entre duas células, maior é a força de atração entre estas, indicando que o percurso entre elas pode ser realizado a velocidades maiores. O cálculo de gradiente está representado nas equações (3) e (4).



- Força atrativa

$$\vec{F}_{att}(q) = -\vec{\nabla}U_{att}(q) = -\vec{\nabla}\left[\frac{1}{2}\xi\rho_{goal}^2(q)\right] \quad (3)$$

- Força repulsiva

$$\vec{F}_{rep,i}(q) = \begin{cases} -\vec{\nabla}U_{rep,i}(q) = -\vec{\nabla}\left[\frac{1}{2}\eta\left(\frac{1}{\rho(q,q_i)} - \frac{1}{\rho_0}\right)^2\right] & \Leftrightarrow \rho(q,q_i) < \rho_0 \\ 0 & \Leftrightarrow \rho(q,q_i) > \rho_0 \end{cases} \quad (4)$$

2.3.4. Método Escolhido

Após a análise dos métodos referidos no estado da arte, concluiu-se que o método dos campos de potencial é o mais adequado para implementar neste projeto. Possui diversas vantagens em relação aos outros métodos, entre elas: não contém limitações no tamanho ou formato dos obstáculos, permite uma fácil aplicação num meio dinâmico e também é o mais simples para implementar no sistema em estudo.





3. Intervenções na Plataforma

Ao longo do desenvolvimento deste trabalho o Atlas-2011 sofreu várias intervenções mecânicas, entre as quais a mudança das rodas, a instalação de um sistema GPS e a instalação de um sistema responsável pela monitorização das baterias. Todos estes serão descritos com mais detalhe seguidamente.

3.1. Mudança de rodas

As rodas do Atlas-2010 apresentavam fortes indícios de desgaste e encontravam-se no limite de utilização, tratavam-se de rodas convencionais de carros de rádio modelismo de escala 1:4 e possuíam cerca de 15cm de diâmetro, uma destas encontra-se ilustrada na Figura 21. A escolha das novas rodas estava condicionada por vários fatores, sendo o mais relevante o diâmetro, que não podia ultrapassar os 23cm de modo a que não fosse preciso efetuar alterações no chassis ou na organização dos elementos mecânicos ou estruturais do robô. Inicialmente previu-se a instalação de um sistema de suspensão no veículo de modo a diminuir as vibrações que o mesmo podia sofrer ao se movimentar em terrenos exteriores. Mas esta tarefa foi contornada com a escolha de rodas pneumáticas de pressão regulável, um sistema deste tipo permite uma locomoção suave mesmo em terrenos mais irregulares. Após muita pesquisa e análise do mercado descobriram-se os candidatos ideais para os requisitos traçados, tratam-se de rodas pneumáticas reguláveis de 8 polegadas de diâmetro, cerca de 20.32 centímetros. Estas foram adquiridas na empresa norte Americana *Harbor Freights*.



Figura 21 - Roda do Atlas-2010



Figura 22 - Roda do Atlas-2011



Como esperado, surgiu a necessidade de desenvolver adaptadores para acoplar estas novas rodas aos veios existentes no robô. Era desejável que estes fossem projetados de modo a permitir, sempre que necessário, a alteração para as rodas antigas de uma maneira simples e rápida. Atingiu-se este objetivo projetando adaptadores para acoplar nas novas rodas, deixando os veios do robô inalterados. Conceberam-se assim, dois tipos de adaptadores diferentes, um para as rodas dianteiras e outro para as traseiras. Este último contém a capacidade de transmitir rotação do veio à roda. Os adaptadores foram desenhados em *software* de modelação 3D CATIA e maquinados, em alumínio, na oficina do Departamento de Engenharia Mecânica.

O adaptador dianteiro é uma peça relativamente simples visto que apenas tem como função a fixação da roda ao eixo, não tendo sido necessário dimensionar rolamentos para esta fixação porque a roda possui rolamentos próprios. Este adaptador consiste essencialmente de um cilindro de duas partes com um furo central passante, como se pode ver na Figura 23. O adaptador traseiro, Figura 25, tem uma complexidade mais elevada, sendo constituído por quatro furos que encaixam no sistema de transmissão de rotação já existente nos veios traseiros. Este sistema consiste em quatro pinos espaçados à mesma distância do centro do veio e a 90° entre eles; esta é a peça fundamental para a transmissão de rotação à roda e encontra-se ilustrada na Figura 24. Esta peça está acoplada por aperto de dois parafusos ao veio traseiro, acompanhando assim a sua rotação. Os rolamentos internos das rodas traseiras foram retirados e o adaptador foi desenhado para ocupar este espaço. Depois de maquinados, os adaptadores traseiros foram inseridos por aperto nas jantes e para garantir que não sofressem qualquer tipo de escorregamento rotacional inseriu-se um parafuso para fixar a jante ao adaptador.



Figura 23 - Adaptador para a roda dianteira (produto final e desenho 3D)

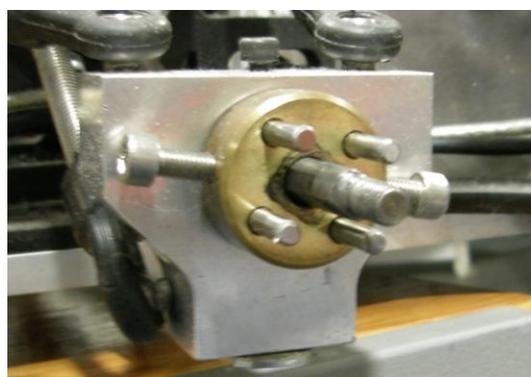


Figura 24 - Sistema de fixação da roda traseira



Figura 25 - Adaptador para a roda traseira (produto final e desenho 3D)

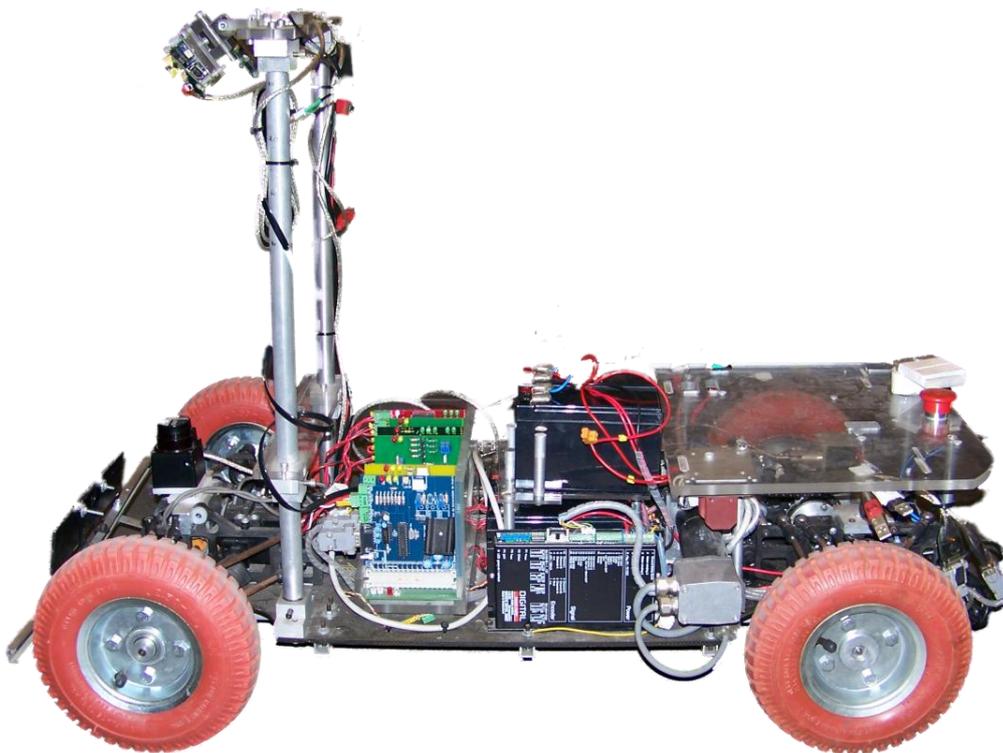


Figura 26 - Atlas 2011 com as rodas novas

Com a instalação final das rodas, obteve-se um ganho de aproximadamente 2.5cm em altura em relação ao solo e os componentes mais inferiores do robô, obtendo uma altura total de 5.5cm, valor suficiente para se deslocar em terrenos mais irregulares. Na Figura 26 encontra-se ilustrado Atlas 2011 com as rodas novas montadas, os desenhos técnicos das peças realizadas encontram-se no Anexo A e no Anexo B.



3.2. Instalação do recetor de GPS

A instalação de um sistema de planeamento de trajetórias por GPS implica, naturalmente, a instalação de um recetor GPS. O recetor usado é um SiRF star III performance *chipset*, ilustrado na Figura 27.

Tabela 3 – Características do recetor GPS [15]

Gerais		Precisão		Tempo de aquisição	
Chipset	SiRF StarIII	Posição	10m (2DRMS) 5m (2D RMSWAAS)	Reaquisição	0.1s
				Hot Start	1s
Frequência	575.42 MHz	Velocidade	0.1 m/s	Warm Start	38s
C/A	1.023 MHz	Tempo	1us sincronizado com o tempo de GPS		
Canais	20				
Sensibilidade	-159 dBm				
Condições dinâmicas		Alimentação		Protocolo	
Altitude	18000 m (Max)	Alimentação	4.5V ~ 6.5V DC input	Nível elétrico	TTL (0V ~ 2.85V)
Velocidade	515 m /s (Max)			Velocidade de transmissão	4800 bps
Aceleração	Menos de 4G	Consumo	44mA	Mensagens de saída	NMEA 0183 GGA, GSA, GSV, RMC, VTG, GLL
<i>Jerk</i>	20m/sec **3				
Características físicas		Temperaturas de funcionamento			
Dimensões	30*30*10.5 ±0.2 (mm)	-40 °C até +85°C			

Como observado na tabela de características, Tabela 3, a comunicação com este recetor é realizada a baixo nível, ao nível TTL (*Transistor-Transistor Logic*). Este sistema usa sinais elétricos baixos, entre 0 e 2.85 Volt, sendo necessário, de modo a permitir a comunicação com um computador, fazer sua conversão. Optou-se pela conversão para níveis elétricos de RS-232, que utiliza sinais entre os 3 e 15 Volt (negativos ou positivos). De modo a atingir estes valores será necessário conceber um circuito de conversão de sinal.

A alimentação deste recetor é realizada a 5 Volt, que pode ser retirada do próprio robô visto que o mesmo dispõe uma placa mãe que fornece 24 Volt constantes. De modo a obter os 5 Volt requeridos será necessário construir um simples circuito elétrico regulador de tensão. Este circuito será implementado numa placa de circuito impresso de modo a acoplar à placa mãe existente no robô. Os sistemas de conversão de sinal e de regulação de tensão serão descritos mais em detalhe no próximo subcapítulo. De modo obter o melhor sinal possível, optou-se por colocar o recetor na



parte mais elevada do robô, como representado na Figura 28, e para a sua fixação ao robô concebeu-se uma peça simples em acrílico, que se encontra ilustrada na Figura 29, os desenhos de definição desta peça encontram-se no Anexo C.

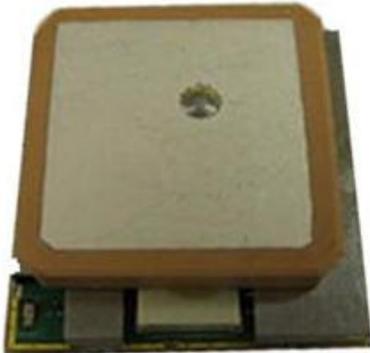


Figura 27 - Recetor GPS SiRF Star III

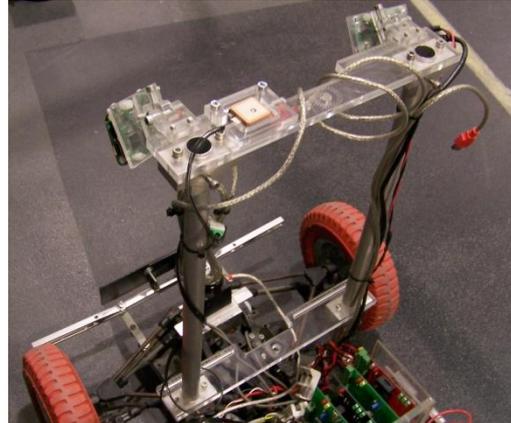
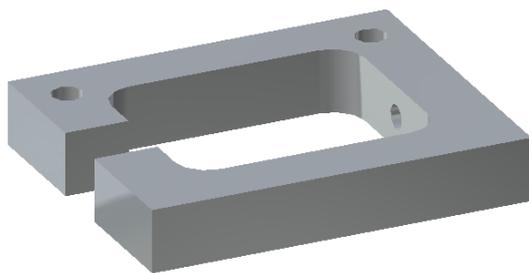


Figura 28 - Localização do recetor GPS no robô



(a)

(a) CAD 3D



(b)

(b) - Produto final

Figura 29 - Suporte para o recetor GPS

3.2.1. Placa para a alimentação e comunicação do GPS

Como mencionado, surgiu a necessidade de criar uma placa de circuito impresso responsável pela alimentação e comunicação do recetor GPS. De modo a obter os 5 Volt necessários para alimentar o recetor optou-se pela utilização do regulador de tensão LM317, que possui a capacidade de regular uma tensão de entrada entre 0 a 40V para uma tensão de saída a variar entre os 1,2V e 37,7V. Esta tensão de saída está dependente das resistências R_1 e R_2 usadas no circuito do regulador, como ilustrado na Figura 30.

A elevação dos sinais elétricos do sensor foi conseguida através da utilização de um circuito integrado conversor de nível Max232, que eleva os sinais de nível TTL para um nível de RS232 para



permitir a comunicação com um computador. A Figura 31 ilustra um esquema de montagem deste circuito

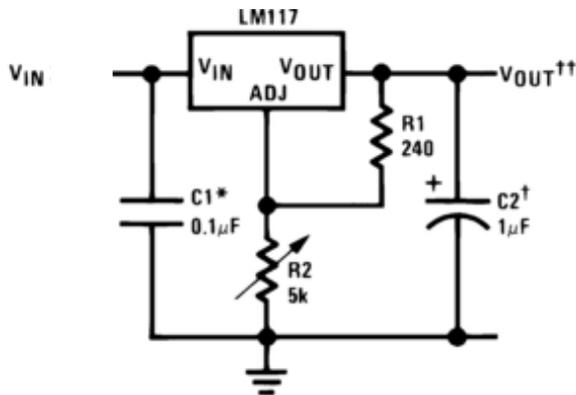


Figura 30 - Regulador de tensão LM317

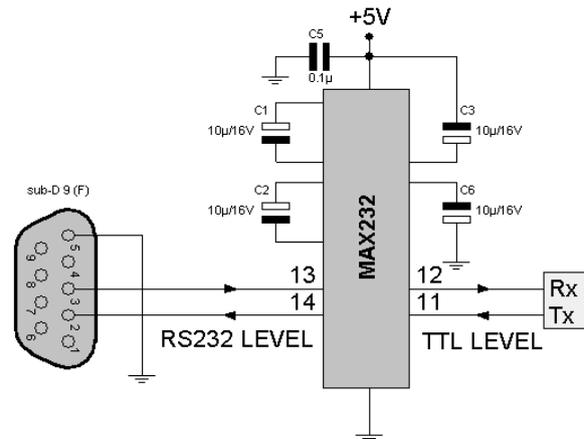
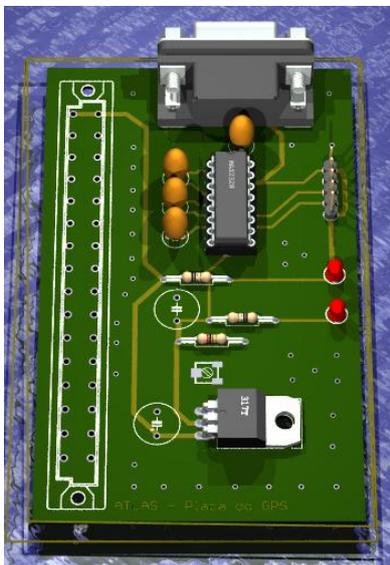


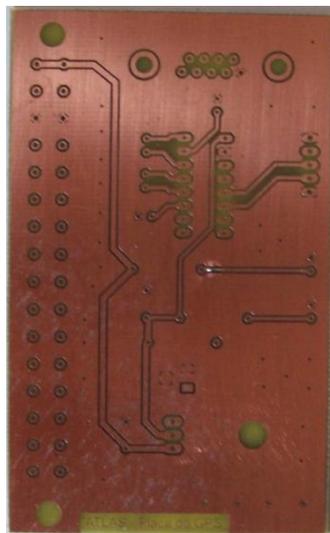
Figura 31 – Conversor de sinal Max232

Em suma, a placa terá de conter os dois sistemas mencionados para conseguir realizar a alimentação e comunicação do recetor. Esta placa foi desenhada no *software* de “construção” de circuitos elétricos *Eagle*. Na Figura 32 podemos comparar a placa 3D gerada pelo *software* com a placa produzida, com e sem os componentes eletrónicos, no Anexo E encontram-se as imagens para a impressão desta placa.



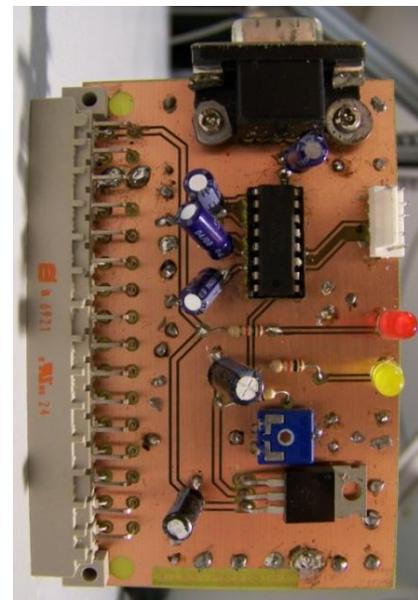
(a)

(a) Pré-visualização (Eagle 3D)



(b)

(b) Placa sem componentes



(c)

(a) Placa final



3.3. Monitorização das baterias

Durante o desenvolvimento deste trabalho surgiu a ideia e necessidade de criar um sistema de monitorização das baterias do robô, para que fosse possível, em qualquer instante, saber os respetivos níveis de carga. Este seria mais um fator a influenciar o planeamento de caminhos em meios exteriores, por exemplo, poderia impedir o robô de se deslocar para uma zona ao qual não teria carga suficiente para depois regressar. O sistema deverá indicar constantemente os níveis das baterias ao utilizador e alerta-lo com uma mensagem de emergência caso alguma bateria atingisse um nível crítico de carga. Após alguma análise do material disponível no mercado, decidiu-se conceber um sistema de raiz. Este teria de possuir a capacidade de ler o nível de carga de cada uma das quatro baterias de 12V, ligadas em série, e transmitir esta informação ao utilizador. Devido a esta ligação em série é impraticável ler diretamente cada bateria de forma independente; assim a leitura é feita em série que poderá levar a valores finais com algum erro de cálculo. Optou-se por utilizar um microcontrolador PIC18F2455 por usufruir a capacidade de, com alguma programação e *hardware* dedicado, comunicar diretamente por USB, permitindo deste modo, a ligação direta a um computador pessoal sem precisar um conversor RS-232-USB. Uma ligação deste tipo permite alimentar o circuito pelo próprio cabo de comunicação, não sendo necessário nenhum sistema adicional para a sua alimentação.

O microcontrolador escolhido possui circuitos integrados capazes de converter sinais analógicos para formato digital, que podem ser utilizados para converter os níveis de carga das baterias num valor digital. O funcionamento base do sistema seria usar as baterias como entradas analógicas no microcontrolador, como ilustrado na Figura 35, onde os valores de carga de cada bateria, em série, são diminuídos de maneira escalar por um divisor resistivo para que possam ser interpretados pelo microcontrolador que, por sua vez, irá convertê-los em valores digitais de fácil leitura e enviá-los por USB ao utilizador. Os microcontroladores possuem uma limitação na alimentação, que deve ser no máximo de 5 Volts, por razões de segurança deve ser assegurado que nenhuma das entradas ultrapasse este valor. O modo mais fácil e seguro de diminuir a valor lido das baterias, de maneira escalar, é com a utilização de divisores resistivos nas ligações de cada entrada. Um divisor resistivo, Figura 34, é um pequeno circuito, constituído por duas resistências, que consegue diminuir o valor de carga de uma maneira escalar.

Os valores convertidos por cada entrada analógica são enviados, ao utilizador, numa mensagem cinco vezes por segundo; estes valores não representam os valores reais da carga das baterias, apenas os valores das conversões analógico-digital efetuadas pelo microcontrolador. A conversão para valores reais de carga apenas é realizada pelo módulo criado, em linguagem C, para a monitorização das baterias, descrito no capítulo 4.2.



Figura 33 - Microcontrolador PIC18F2455

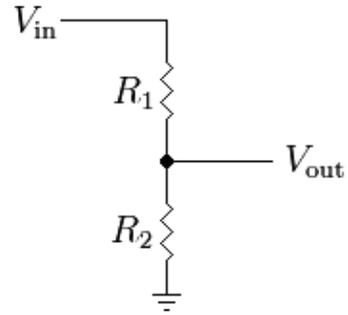


Figura 34 - Divisor resistivo

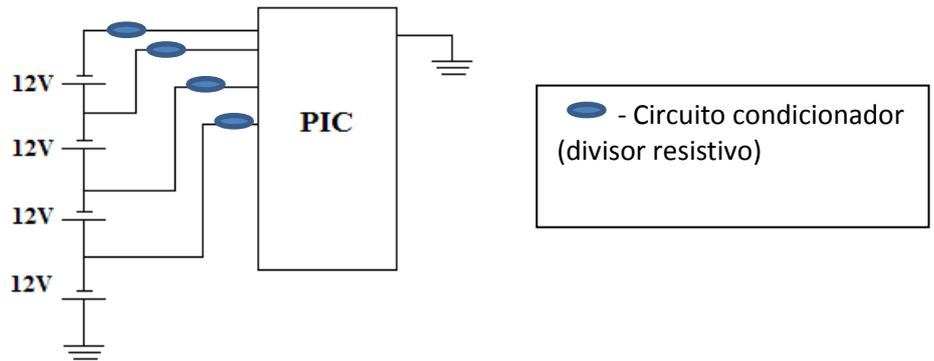


Figura 35 - Esquema das entradas do microcontrolador

3.3.1. Placa de circuito impresso para a monitorização das baterias

O circuito responsável pela monitorização das baterias encontra-se ilustrado na Figura 36; é constituído por quatro divisores resistivos para diminuir a tensão medida de cada bateria, um microcontrolador, um led de sinalização, um cristal de 20 MHz e quatro condensadores, essenciais para que a comunicação possa ser feita por USB.

Dimensionamento das resistências dos divisores resistivos:

Atendendo à nomenclatura dada na Figura 34, as resistências utilizadas são dimensionadas pela equação (5) [18]. Por razões de segurança, estes valores foram calculados com uma margem de segurança de 15% no valor de tensão de entrada, de modo a que a corrente de saída do divisor resistivo nunca ultrapasse os 5 Volt. De seguida se apresentaram os cálculos realizados para o



dimensionamento das resistências utilizadas em cada um dos quatro divisores resistivos.

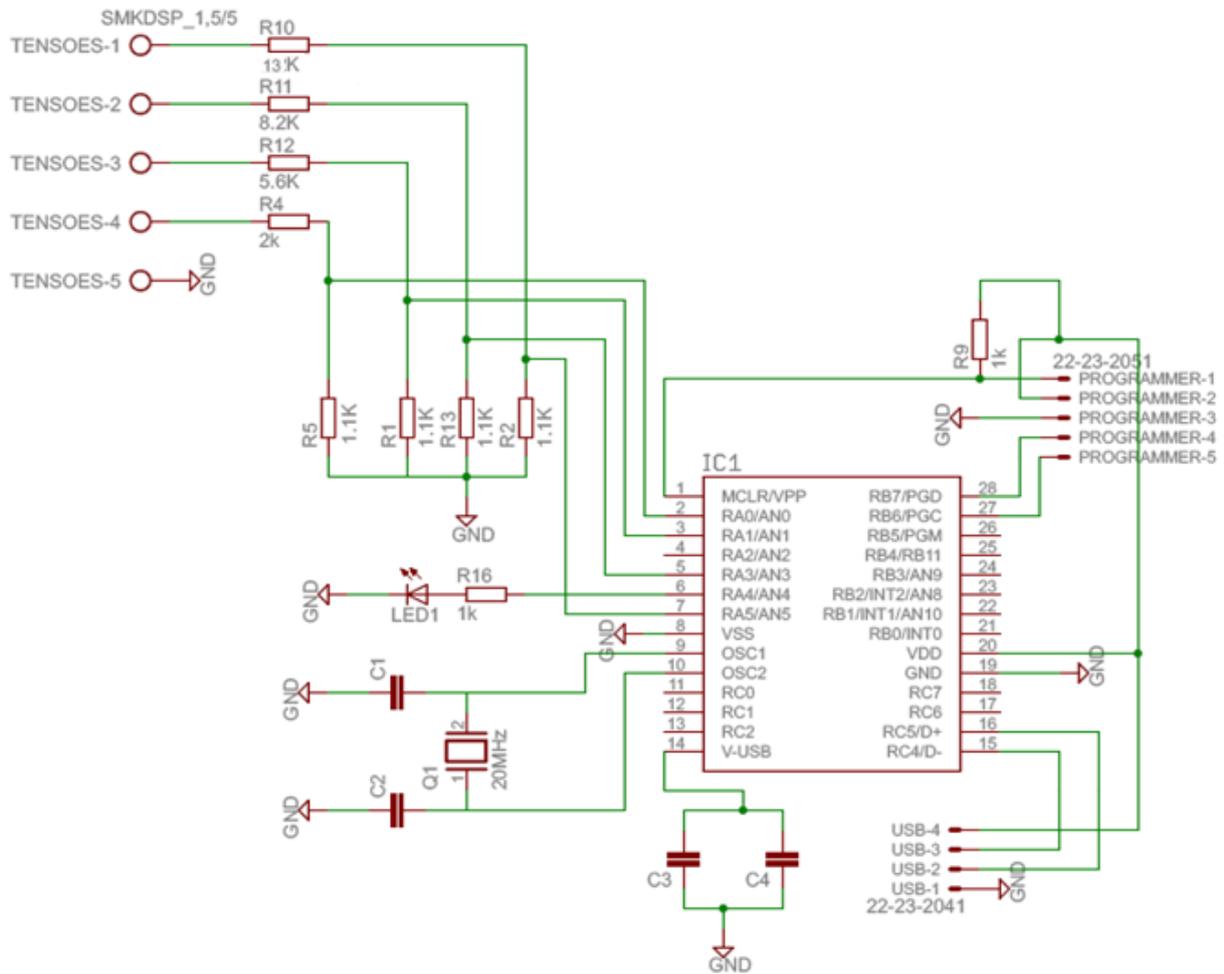


Figura 36 - Circuito da monitorização das baterias

$$V_{out} = \frac{R_2}{R_1 + R_2} \cdot V_{in} \quad (5)$$

- 12V

$$5 = \frac{R_2}{R_1 + R_2} * 12 * 15\% \Leftrightarrow \frac{5}{13.8} = \frac{R_2}{R_1 + R_2} \Leftrightarrow$$

$$\Leftrightarrow 5(R_1 + R_2) = 13.8 * R_2 \Leftrightarrow 5R_1 = 8.8 * R_2$$

$$\Leftrightarrow R_1 = \frac{8.8}{5} * R_2 \approx 1.76R_2$$

- 24V

$$5 = \frac{R_2}{R_1 + R_2} * 24 * 15\% \Leftrightarrow R_1 = \frac{22.6}{5} * R_2 \approx 4.52R_2$$



- 36V

$$5 = \frac{R_2}{R_1 + R_2} * 36 * 15\% \Leftrightarrow R_1 = \frac{36.4}{5} * R_2 \approx 7.28R_2$$

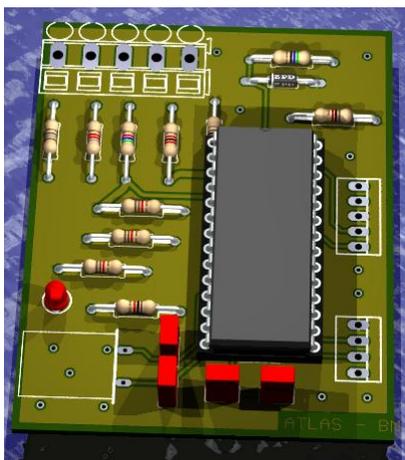
- 48V

$$5 = \frac{R_2}{R_1 + R_2} * 48 * 15\% \Leftrightarrow R_1 = \frac{50.2}{5} * R_2 \approx 10.04R_2$$

Na Tabela 4 encontram-se os valores das resistências escolhidas para os divisores resistivos. Após esta escolha e dos restantes componentes do circuito, a placa foi devidamente montada e soldada. Na Figura 37 encontram-se as várias etapas do processo de realização desta placa, desde a sua conceção em 3D até à montagem dos componentes do circuito, no Anexo D encontram-se as imagens para a impressão desta placa.

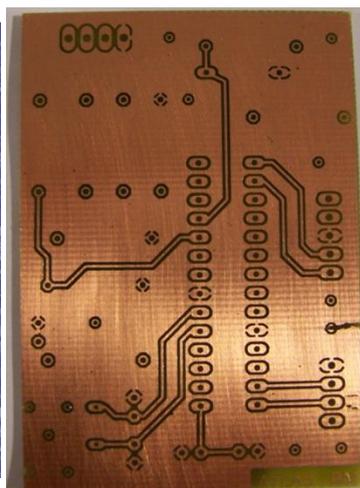
Tabela 4 - Resistências utilizadas nos divisores resistivos

Tensão (Volts)	Relação	R ₁ (kΩ)	R ₂ (kΩ)
12	$R_1 = \frac{8.8}{5} * R_2 \approx 1.76R_2$	1.1	2
24	$R_1 = \frac{22.6}{5} * R_2 \approx 4.52R_2$	1.1	5.6
36	$R_1 = \frac{36.4}{5} * R_2 \approx 7.28R_2$	1.1	8.2
48	$R_1 = \frac{50.2}{5} * R_2 \approx 10.04R_2$	1.1	13



(a)

(a) -Pré-visualização (Eagle 3D)



(b)

(b) - Placa sem componentes



(c)

(c) – Placa final



4. Sistemas Desenvolvidos

Todo o sistema existente no ATLAS 2010 foi alterado para um sistema modular seguindo as regras da filosofia CARMEN/IPC. Neste capítulo serão descritas estas alterações bem como os módulos criados e os tipos de mensagens que utilizam. Serão ainda detalhados os vários módulos criados para a monitorização das baterias e para navegação autónoma em meios exteriores por GPS.

4.1. Migração de todo o *software* para a filosofia CARMEN/IPC

Vários módulos foram desenvolvidos no processo de migração da arquitetura antiga para nova, a modular. Estes módulos podem ser divididos em três categorias: interface de *hardware*, processamento de informação e planeamento e decisão. A primeira é a responsável por todas as interações com o *hardware*, sendo estas de aquisição de dados ou controlo dos motores. Os módulos de processamento de informação são os responsáveis pelo processamento dos dados adquiridos, em que cada módulo é apenas responsável por um tipo de processamento, como a deteção de obstáculos ou da estrada, por exemplo. A última categoria é dedicada às capacidades de decisão e escolha do robô, em que os resultados do processamento dos dados são analisados e, dependendo do contexto, é decidido o comportamento do robô.

O módulo servidor que se encarrega da distribuição da informação pelos restantes módulos denomina-se de módulo central, sendo o módulo mais importante e que deve ser iniciado antes de todos os restantes. Na Figura 38 encontra-se uma lista dos módulos criados para a condução autónoma *indoor* do ATLAS-2011, esta imagem representa o programa de *process control* utilizado nesta condução, que torna o funcionamento mais *user-friendly* e permite um arranque rápido de todos os módulos necessários. O módulo denominado de *ParamDaemon2000* é o responsável pela definição de todos os parâmetros variáveis do sistema, como por exemplo, as características das câmaras e as limitações da velocidade. Este módulo lê um ficheiro de texto onde todos estes parâmetros estão previamente definidos, e a alteração de qualquer um deles, desde a calibração das câmaras até os parâmetros dos motores, pode ser facilmente realizada a partir deste ficheiro. De modo a não tornar este subcapítulo muito repetitivo apenas serão abordados, com mais detalhe, os módulos de interface com o *hardware*, visto que os restantes módulos contêm programas



complexos, e extensos, que não foram desenvolvidos durante a realização deste trabalho, apenas as suas migrações para módulos individuais. Todos os módulos contêm um funcionamento idêntico, baseando-se numa troca de informação através da publicação e subscrição de mensagens pelos métodos descritos no capítulo 2.1. Toda a migração foi um processo delicado que contou com a participação da equipa Atlas.



Figura 38 - Módulos criados para a condução autónoma *indoor*

Módulos de interface com o hardware:

➤ Módulo de base do robô

Este módulo é o responsável pela interface com os motores de tração e direção, e também efetua a leitura de entradas e saídas digitais como controlo das luzes e do travão. Na Figura 39 encontra-se uma ilustração explicativa da troca de informação deste módulo, que recebe a mensagem *lar_atlas_dir_and_speed_message*, demonstrada na Estrutura 2, e o traduz para o *hardware* específico do robô, definindo a velocidade e direção pedida pela mensagem. Desta maneira, diferentes robôs com diferentes módulos de base são capazes de receber a mesma mensagem de comando. O ATLAS MV e o ATLAS 2000 possuem bases muito diferentes mas as mensagens que recebem são as mesmas; só o método de processamento destas é que varia. Este



módulo publica uma mensagem contendo informação sobre o estado atual do robô; esta mensagem encontra-se detalhada na Estrutura 3.

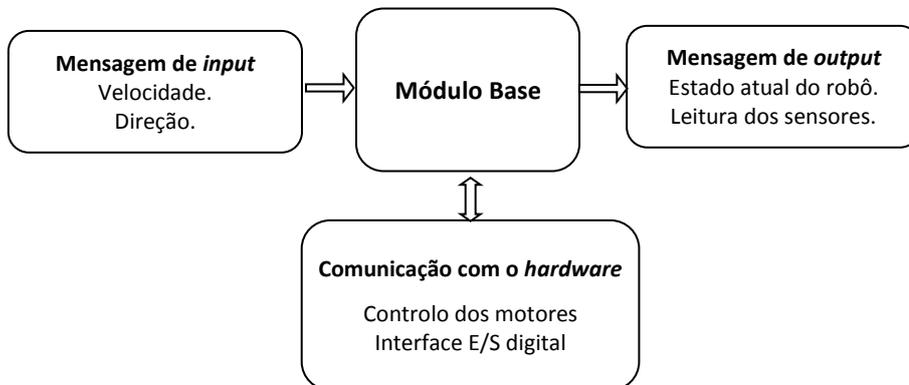


Figura 39 - Módulo da Base

```

Typedef struct
{
double dir;           (direção)
int speed;           (velocidade)
double timestamp;    (timestamp da msg)
char *host;          (usado pelo IPC)
} lar_atlas_dir_and_speed_message;
    
```

Estrutura 2 - Mensagem recebida pelo Base

```

Typedef struct
{
double dir;           (direção atual)
int speed;           (velocidade)
double distance_traveled (distância percorrida)
double timestamp;    (timestamp da msg)
char *host;          (usado pelo IPC)
} robot_status_message
    
```

Estrutura 3 - Mensagem publicada pelo Base (estado do robô)

➤ Módulo de aquisição das câmaras

O módulo de aquisição das câmaras, esquematizado na Figura 40, tem como objetivo adquirir imagens das câmaras *Firewire* instaladas nos robôs, corrigi-las e publica-las na mensagem deste módulo (Estrutura 4). Todo o intercâmbio de mensagens é seguido pelos métodos já referidos, publicação/subscrição, pedido/resposta e com memória partilhada. Este módulo permite ainda a definição dos parâmetros das câmaras como brilho, saturação, equilíbrio de brancos, obturador e outros podem ser definidos para o valor desejado.

➤ Módulo de aquisição do laser

Este módulo, ilustrado na Figura 41, apenas possui a tarefa de adquirir os dados laser e envia-los para os módulos interessados, a troca de mensagens é realizada pelos três métodos descritos anteriormente. Durante a sua inicialização, o módulo envia uma mensagem de configuração para o laser, definindo os seus parâmetros. As mensagens enviadas pelo laser contêm informação dos



pontos detetados pelo mesmo, a sua mensagem de configuração e de informação encontra-se detalhada na Estrutura 5.

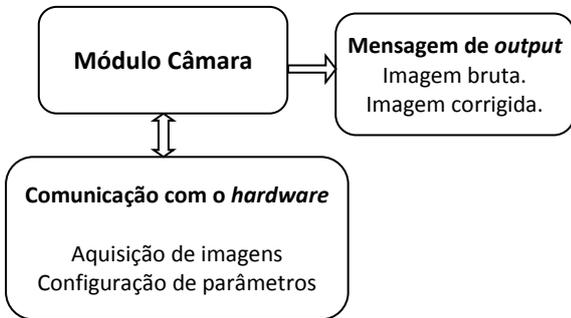


Figura 40 - Módulo das câmaras

```

Typedef struct {
    int width;           (largura em pixels)
    int height;         (altura em pixels)
    int bytes_per_pixel; (3-> RGB)
    int image_size;     (w*h*3)
    char *image;        (ponteiro para a imagem)
    double timestamp;   (timestamp da mensagem)
    char *host;         (usado pelo IPC)
} lar_ubcamera_image_message;
    
```

Estrutura 4 - Mensagem dos módulos de aquisição de imagem

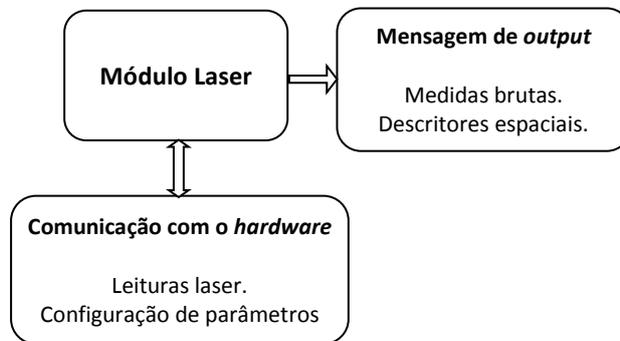


Figura 41 - Módulo laser

<pre> Typedef struct { carmen_laser_laser_type_t laser_type (modelo do laser) double start_angle; (ângulo do primeiro feixe) double fov; (field of view – campo de visão do laser) double angular_resolution (resolução angular, até 0.25) double maximum_range; (alcance máximo, 30m) double accuracy; (precisão, 0.1m) carmen_laser_remission_type_t remission_mode (não usado) } lar_laser_laser_config_t; </pre>	<pre> Typedef struct { int id; (id do laser) carmen_laser_laser_config_t config (modelo do laser) int num_readings; (nº de pontos lidos) float *range; (distância) int num_remissions; (nº de remissões) float *remission; (valor da remissão) double timestamp; (timestamp da mensagem) char *host; (usado pelo IPC) } lar_laser_laser_message; </pre>
---	---

Estrutura 5 - Mensagens do laser

➤ Módulo de controlo remoto

Este módulo permite o controlo remoto do *hardware* do robô utilizando um *gamepad* convencional, e não possui quaisquer rotinas próprias de publicação/subscrição. Simplesmente utiliza as rotinas implementadas em cada módulo que quer controlar. Por exemplo, no caso da velocidade e direção, utiliza as rotinas implementadas no módulo de base do robô para fazer o seu controlo. O módulo em questão possui dois modos de operações distintas: um modo de controlo e



um de tutorial. O primeiro é capaz de controlar a velocidade, direção e luzes, publicando a mensagem *lar_atlas_dir_and_speed_message* (Estrutura 2). No modo tutorial o acionamento de botões ou de eixos fornecerá um tutorial audio; isto é conseguido através do envio de instruções ao módulo de reprodução de som, em que cada botão é previamente associado com um tutorial áudio.

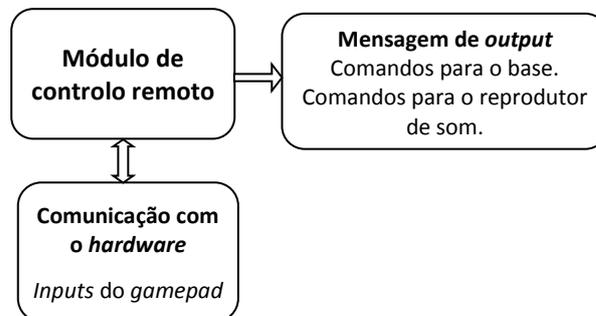


Figura 42 - Módulo de controlo remoto

➤ Módulo de reprodução de som

A reprodução de som é realizada por este módulo que possui a capacidade de reproduzir faixas de audio previamente gravadas; por razões de sincronização também consegue alertar outros módulos quando está ocupado ao reproduzir som. Qualquer módulo pode requerer a reprodução de uma mensagem de som, necessitando apenas de publicar a mensagem *lar_soundplayer_message*. Esta mensagem está definida na Estrutura 6, e é constituída pela identificação da faixa a ser reproduzida, podendo ser pelo nome do ficheiro (em *string*) ou da sua identificação numérica. Após a receção do comando, o reproduztor utiliza funções da biblioteca *Libao* [16] para reproduzir o audio desejado. O módulo de reprodução ainda indica o seu estado, ocupado ou disponível, publicando a *lar_soundplayer_status_message*, definida na Estrutura 7. Este módulo permite uma interatividade utilizador/robô melhorada e oferece ainda facilidades de *debug*.

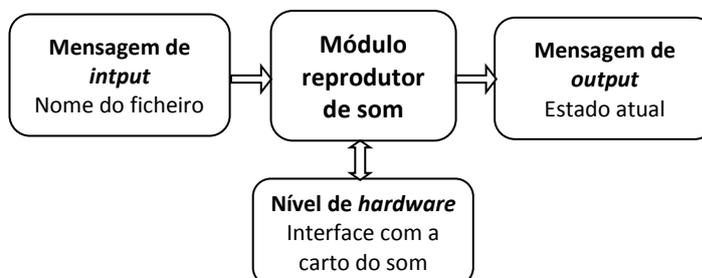


Figura 43 - Módulo reproduztor de som



```
Typedef struct {  
  int filenumber;      (identificação numérica)  
  char *filename;     (nome do ficheiro)  
  int mode;           (modo de id, por nº ou nome)  
  double timestamp;  (timestamp da mensagem)  
  char *host;         (usado pelo IPC)  
} lar_soundplayer_message;
```

Estrutura 6 - Mensagem de comando para o reproduztor

```
Typedef struct {  
  int status;          (estado  
do reproduztor)  
  double timestamp;  (timestamp  
da mensagem)  
  char *host;         (usado  
pelo IPC)  
} lar_soundplayer_status_message;
```

Estrutura 7 - Mensagem do estado do reproduztor

➤ Módulo da fusão de sensores (*IPM Module*)

Representado na Figura 44, este módulo é o responsável pela fusão da informação proveniente das câmaras e do sensor laser instalado no robô. Recria uma representação comum às medidas tiradas, sendo estas imagens ou varrimentos do laser, ou seja, recria informação com todos os dados sobrepostos. Este módulo consegue fundir várias imagens, capturadas das múltiplas câmaras em conjunto com a informação laser, gerando imagens aumentadas da estrada, numa perspetiva *birdview*. Este algoritmo está descrito em mais detalhe num artigo publicado por alguns elementos da equipa Atlas [17]. A Figura 45 demonstra a imagem *birdview* obtida pela fusão de imagens de duas câmaras diferentes; esta perspetiva oferece mais facilidades na deteção da estrada. A imagem e informação dos pontos laser detetados são encaminhadas na mensagem publicada por este módulo, como representado na Estrutura 8. Durante a sua inicialização é realizada a leitura dos parâmetros que definem a área de interesse, região retangular na frente do robô que será analisada na deteção de estrada. Se por alguma razão um sensor particular armazenar informação de uma zona de não interesse para o robô, fora da região referida, este módulo assegura que esta informação, sendo classificada como desprezável, seja retirada da informação fundida. O objetivo principal desta tarefa é procurar uma representação comum aos vários tipos de sensores e/ou configurações. Assegurando que, independentemente da configuração sensorial do robô, a informação é reestruturada num referencial comum. A vantagem é que os módulos de processamento (como extração de características, deteção de estradas e obstáculos, etc.) podem confiar numa informação sincronizada, não sendo necessário qualquer tipo de reconfiguração posterior para os comparar.



Figura 44 - Módulo do IPM

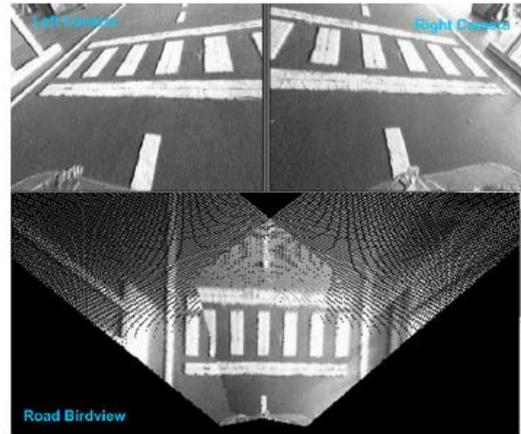


Figura 45 - Exemplo de fusão de imagens de duas câmaras para perspectiva *birdview*

```
typedef struct {
    int width;           (largura da imagem em pixels)
    int height;         (altura da imagem em pixels)
    int bytes_per_pixel; (usualmente 3, RGB)
    int image_size;     (tamanho)
    char *image;
    int numlaserpts;    (nº de pontos laser)
    int *xpts;          (coord. x dos pontos laser, em mm)
    int *ypts;          (coord. y dos pontos laser, em mm)
    double rawdata_timestamp; (timestamp dos dados brutos)
    double processing_time;  (tempo de processamento)
    double timestamp;
    char *host;
}ipm_data_message;
```

Estrutura 8 - Mensagens do módulo de fusão

Módulos de Processamento de informação:

Estes são os módulos responsáveis por todo o pós processamento dos dados obtidos pelos módulos de comunicação com o *hardware*. Encarregam-se de filtrar os dados para obter a informação necessária para que o robô possa tomar uma decisão de trajetória. As suas principais funções englobam a deteção da estrada, dos semáforos, da passadeira e do parque de estacionamento, os resultados destes processamentos são publicadas em mensagens separadas. Estes módulos utilizam algoritmos complexos criados pela equipa Atlas e que apenas serão mencionados de seguida:

- Módulo de Deteção de semáforos (*tlhfcs – Traffic Lights detection using HaarFeatures and Color Detection*)
- Módulo de Deteção de estrada (*HCLR – Hough Circles/lines Road detecting*)



- Módulo de Correção da Percepção (*PCDR – Perception Correction using Dead Reckoning*)
- Módulo de Detecção da Passadeira (*CWPC – CrossWalk detection With Pixel Count*)
- Módulo do Parque de Estacionamento (*PDTM – Parking space Detection algorithm using Template Matching*)

Módulos Planeamento e Decisão:

Todo o planeamento de trajetórias é efetuada por um módulo, o *MotionPlanner*, que recebe as mensagens dos módulos de processamento de informação e possui a capacidade de escolher uma trajetória segura para o robô. Durante a competição de condução autónoma o robô utiliza um módulo de máquina de estados, que identifica a situação em que o robô se encontra e realiza o seu planeamento consoante o seu estado, podendo se encontrar na fase de partida, corrida normal ou deteção de semáforos por exemplo. Mais especificamente estes módulos são:

- Módulo do planeamento de trajetória (*MotionPlanner – using Composite Trajectories*)
- Módulo de Máquina de Estados (*Robotica_AD – Context manager for the autonomous driving competition*)

4.2. Monitorização das baterias

Após o desenvolvimento do *hardware* para a leitura dos níveis das baterias, referido no capítulo 3.3, desenvolveu-se o módulo dedicado para esta monitorização. Este módulo é responsável pela comunicação com o *hardware*, interpretação dos dados recebidos, sua conversão em valores reais de carga e finalmente, pela publicação na mensagem *bm_status_message*, que se encontra detalhada na **Estrutura 9**. Os dados recebidos por este módulo são os valores das conversões analógicas/digitais efetuadas pelo microcontrolador utilizado no circuito de monitorização de baterias. Sendo necessário realizar uma conversão destes dados de modo a obter os valores reais de carga, conseguida com a utilização de uma escala obtida experimentalmente. A escala foi obtida através da análise dos dados enviados pelo microcontrolador e dos valores reais de carga das baterias, obtendo-se uma relação linear entre eles. Após a conversão para valores reais das baterias os dados são publicados na mensagem *bm_status_message*. A monitorização está constantemente a ser realizada e representada na consola do programa, como ilustrado na Figura 46. Por enquanto esta informação não entra no planeamento de caminhos do robô, mas futuramente poderá ter influência nesse planeamento.



```
eugenio@ubuntu: ~/lar/src/sensors/battery_monitoring
File Edit View Terminal Help
eugenio@ubuntu:~/lar/src/sensors/battery_monitoring$ ./bm
modulereadcommandlineparameters: reading execution flags
modulereadcommandlineparameters: done
CARMEN - Carnegie Mellon Robot Navigation Toolkit - Version 0.7.0

ipc_name:bm-1964

==== Battery Monitoring activated ====

battery 1->12.650000
battery 2->12.830000
battery 3->13.010000
battery 4->12.780000

battery 1->12.650000
battery 2->12.830000
battery 3->13.010000
battery 4->12.780000

battery 1->12.650000
battery 2->12.830000
battery 3->13.010000
```

Figura 46 - Printscreen da monitorização das baterias

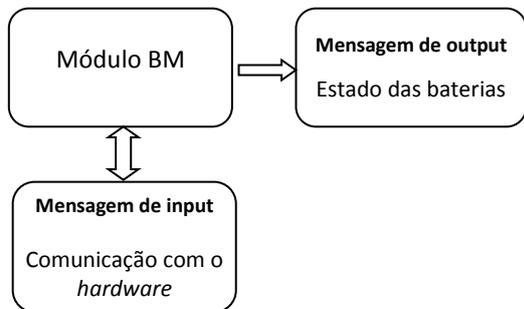


Figura 47 - Módulo de monitorização das baterias

```
Typedef struct {
double bat1;      (estado da bateria nº1)
double bat2;      (estado da bateria nº2)
double bat3;      (estado da bateria nº3)
double bat4;      (estado da bateria nº4)
double timestamp; (timestamp da mensagem)
char *host;       (usado pelo IPC)
} bm_status_message;
```

Estrutura 9 - Mensagem da monitorização das baterias

4.3. Navegação por GPS

Neste subcapítulo serão abordados os módulos criados para auxiliar a navegação por GPS, tendo sido implementada a técnica de planeamento por campos de potencial, sendo a mais interessante das técnicas abordadas. Ante da criação do *software* em linguagem C, desenvolveu-se um programa simples em *Matlab* para melhor compreensão dos conceitos desta técnica. Este é capaz de resolver problemas simples de planeamento de trajetórias num espaço bidimensional com obstáculos estáticos pré-definidos, na Figura 48 encontra-se a ilustração de um problema genérico bem como a representação dos valores dos potenciais calculados. A escolha da trajetória é realizada através da análise das células vizinhas da posição atual do robô, sendo escolhida a célula contendo o potencial mais baixo como o próximo destino local do robô. No arranque do programa é criada uma matriz da dimensão adequada ao espaço em análise, em que cada elemento da matriz representa uma célula do espaço. Após o cálculo do potencial atrator e repulsor, os dois são somados de modo a obter o campo de potencial global total, permitindo assim a análise célula a célula de modo a obter a



trajetória até ao ponto destino. A realização deste programa permitiu uma melhor compreensão do método e auxiliou no desenvolvimento do programa em C.

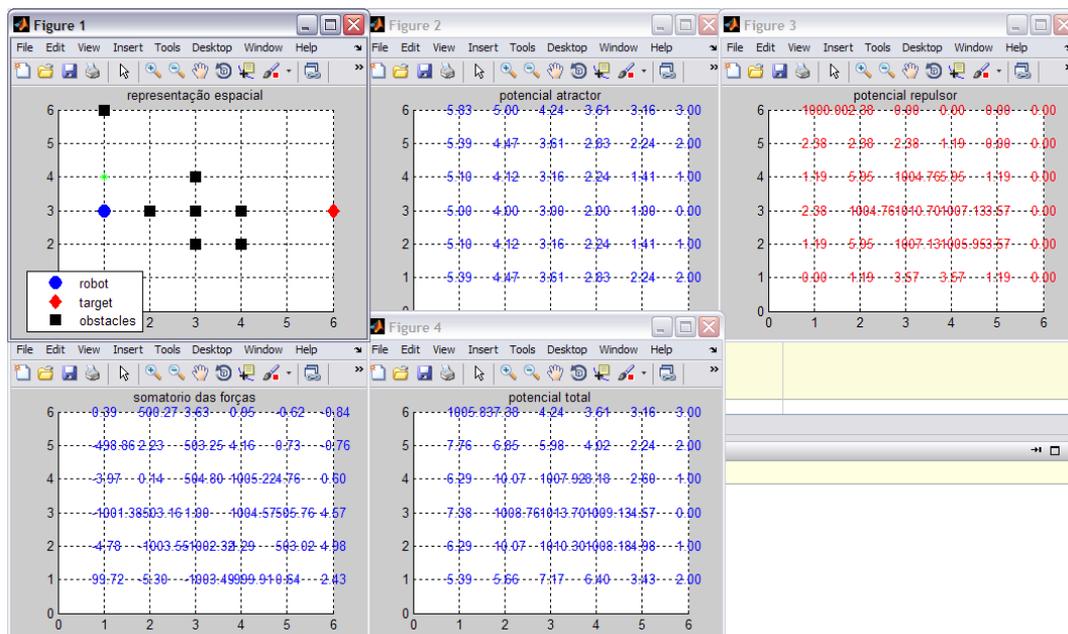


Figura 48 - Campos de Potencial – *printscreen* do programa realizado em Matlab

4.3.1. Comunicação com o recetor GPS

Criou-se um módulo, esquematizado na Figura 49, dedicado à comunicação com recetor GPS, que comunica com o *hardware* desenvolvido no capítulo 3.2. As funcionalidades deste módulo incluem o estabelecimento da comunicação com recetor, a separação dos dados recebidos e a publicação das mensagens devidas. A comunicação é realizada através de uma porta de série a uma velocidade de transmissão de 4800 bits/s. Os dados recebidos vêm no formato especificado na Tabela 2, decidiu-se colocar cada uma das frases referidas numa mensagem específica com cada tipo de dado guardado numa variável separada. A estrutura de uma destas frases encontra-se discriminada na Estrutura 10.

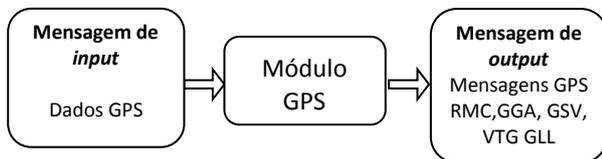


Figura 49 - Módulo GPS

```

Typedef struct {
  int nr;           (nº da unidade de GPS)
  int validity;    (validade)
  double utc;      (Universal Time Coordinated)
  double latitude; (latitude como nº flutuante)
  double latitude_dm; (latitude no formato DDMM.MMMM)
  char lat_orient; (Norte,N, ou Sul,S)
  double longitude; (long. como nº flutuante)
  double longitude_dm; (long. no formato DDMM.MMMM)
  char long_orient; (Este,E, ou Oeste,W)
  double speed;     (velocidade em m/s)
  double true_course; (alinhamento com N em rad)
  double variation; (variação magnetica)
  double timestamp; (timestamp da mensagem)
  char*      (host);
} gps_gprmc_data_message;
  
```

Estrutura 10 - Mensagem GPS-RMC

4.3.2. GPS Mapping

A representação no mapa, ou *mapping*, da localização do robô é efetuada através deste módulo, que não publica qualquer mensagem e apenas subscreve as mensagens do módulo de GPS. Além da representação, o módulo também permite efetuar um registo dos dados GPS, guardando a informação da posição num ficheiro de texto e o mapa com o trajeto efetuado num ficheiro de imagem. Ao inicializar, o programa pergunta ao utilizador em que zona este se encontra e confirma-o através de coordenadas enviadas pelo GPS; o programa apenas prossegue se o utilizador escolher a zona em que se encontra. O programa foi realizado de modo que a adição de novas zonas fosse uma tarefa simples, sendo apenas necessário uma imagem da zona e calibrar alguns parâmetros do algoritmo. Ao seleccionar a zona correta é carregado o respetivo mapa e inicializado o mapeamento da trajetória. A localização do robô no mapa é conseguida através de um algoritmo relativamente simples que converte a posição real, latitude e longitude, na posição na imagem, coordenadas em *pixels*, a conversão é conseguida através da utilização das equações (6), (7), (8) e (9). As conversões são realizadas tendo em conta que o eixo x da imagem está alinhado com o eixo longitudinal e o eixo y com o eixo latitudinal. O algoritmo representa a posição do robô até o programa ser terminado, na Figura 50 encontra-se um exemplo de registo de posição através deste módulo na zona da reitoria da Universidade de Aveiro.



$$Lat_{conv} = \frac{\Delta Lat}{\Delta y_{pix}} \quad (6)$$

$$Lon_{conv} = \frac{\Delta Lon}{\Delta x_{pix}} \quad (7)$$

$$y = \frac{(Lat_{start} - Latitude)}{Lat_{conv}} \quad (8)$$

$$x = \frac{(Lon_{start} - Longitude)}{Lon_{conv}} \quad (9)$$

Legenda:

Lat_{conv} – Constante de conversão de latitude;

Lon_{conv} – Constante de conversão de longitude;

ΔLat – Intervalo de coordenadas latitudinais do mapa na imagem;

ΔLon – Intervalo de coordenadas longitudinais do mapa na imagem;

Δy_{pix} – Altura da imagem, em pixels;

Δx_{pix} – Comprimento da imagem, em pixels;

Lat_{start} – Latitude do ponto inicial da imagem (Coordenada de referência);

Lon_{start} – Longitude do ponto inicial da imagem (Coordenada de referência);

Latitude – Latitude atual;

Longitude – Longitude atual;

y – Coordenada vertical do pixel;

x – Coordenada horizontal do pixel;

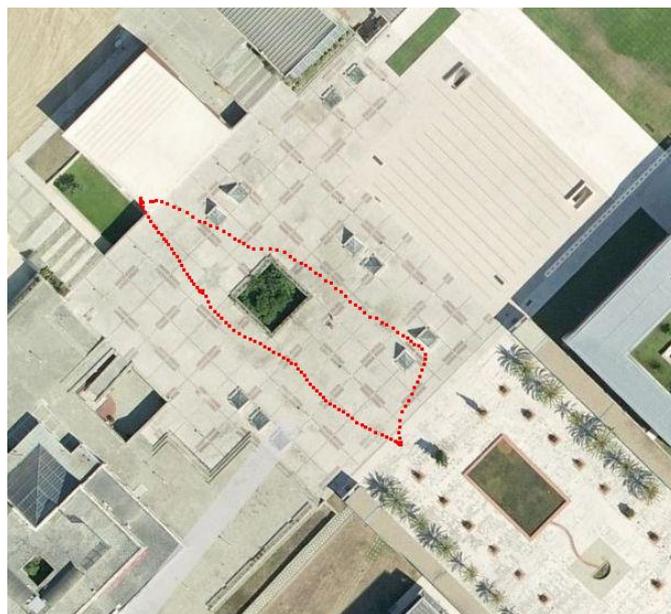


Figura 50 - GPS Mapping - Exemplo

4.3.3. Obtenção da Orientação

A obtenção de uma orientação válida foi um dos grandes problemas encontrados durante a realização deste projeto e sendo um componente essencial no planeamento de caminhos, é de extrema importância que seja a mais correta possível. Com o sistema atual, a orientação provém da informação enviada pelo GPS, sendo calculada através da diferença de coordenadas entre dois pontos percorridos. Devido ao erro de leitura de posição do GPS, o valor calculado de orientação



pode conter incertezas. Outro problema com este sistema é a impossibilidade de obter uma orientação válida com o recetor imobilizado, sendo necessário criar uma sub-rotina para obter uma orientação válida durante o arranque do sistema, que será descrita no subcapítulo 4.3.4. O módulo da obtenção da orientação encontra-se subscrito às mensagens *gps_gprmc_data_message* e *robot_status_message* e apenas publica uma mensagem que contém a orientação válida. Na Figura 51 encontra-se ilustrada a comunicação deste módulo e na Estrutura 11 o tipo de mensagem que publica. O valor da orientação obtida pelo recetor apenas é válido quando o mesmo se encontra em movimento; ao se encontrar parado, devido ao erro de leitura de posição, o seu valor poderá estar incorreto. De modo a corrigir este erro, as orientações obtidas são guardadas em memória e é efetuada uma verificação constante da velocidade do robô através da mensagem *robot_status_message*. Esta verificação constante permite averiguar quando este se encontra imobilizado, logo, quando o GPS regista uma orientação pouco fiável. Quando tal acontece é publicada a última orientação obtida quando este ainda se encontrava em movimento, voltando apenas a publicar a orientação recebida pelo recetor quando voltar a se encontrar em movimento. A mensagem publicada deste módulo consiste em duas variáveis, uma indicar se a orientação encaminhada pelo módulo GPS é válida e outra com a orientação válida. O módulo foi desenvolvido de modo a que, num futuro próximo, a inserção de um sensor dedicado à obtenção da orientação, como uma bússola digital, se realize de uma maneira simples.



Figura 51 - Módulo Orientação

```

typedef struct {
  int orient_ok; (validação da orientação)
  double orient; (orientação)
  double timestamp;
  char *host;
}orient_data_message;
  
```

Estrutura 11 - Mensagem de orientação

4.3.4. Navegação Inicial para a obtenção de uma orientação válida

Como referido, o recetor apenas consegue obter uma orientação válida depois de se encontrar em movimento, que é uma situação que não se aplica no arranque do sistema de condução autónoma. Assim, necessitou-se de conceber um pequeno módulo, ilustrado na Figura 52, responsável pela aquisição de uma orientação válida na fase inicial do planeamento de caminhos. O módulo envia uma mensagem ao módulo de base do robô, dando ordem para andar em linha reta de modo a que o recetor GPS possa obter uma orientação viável. O robô apenas se desloca se tiver



espaço para o fazer, o módulo de navegação inicial encontra-se subscrito às mensagens do laser, conseguindo assim verificar a existência ou não de obstáculos em frente ao robô. Caso não haja obstáculos, o módulo envia uma mensagem para o módulo base a indicar que o caminho se encontra livre e que pode andar em frente. Durante este deslocamento são efetuadas leituras da orientação indicada pelo GPS, e caso esta se mantenha com pouca variância durante algumas leituras considera-se que se obteve uma orientação válida. Considerou-se o número de leituras a analisar e a variância máxima admissível de 10 e 15% respetivamente, a alteração destes pode facilmente ser realizada a partir do código. Caso algum obstáculo apareça diante ao robô, este imobiliza-se de imediato, voltando apenas a se deslocar quando não encontrar nenhum obstáculo na sua frente. Ao obter uma orientação válida o módulo publica a sua mensagem, detalhada na Estrutura 12, com a orientação obtida e termina a sua execução. O processo de funcionamento do módulo em análise encontra-se esquematizado no diagrama apresentado na Figura 53.

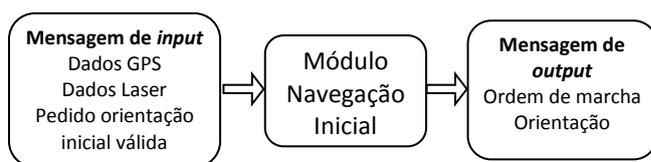


Figura 52 - Módulo de Navegação Inicial

```
typedef struct {  
    int move; (ordem para andar)  
    double orient; (orientação obtida)  
    double timestamp;  
    char *host;  
}orient_data_message;
```

Estrutura 12 - Mensagem de Navegação Inicial

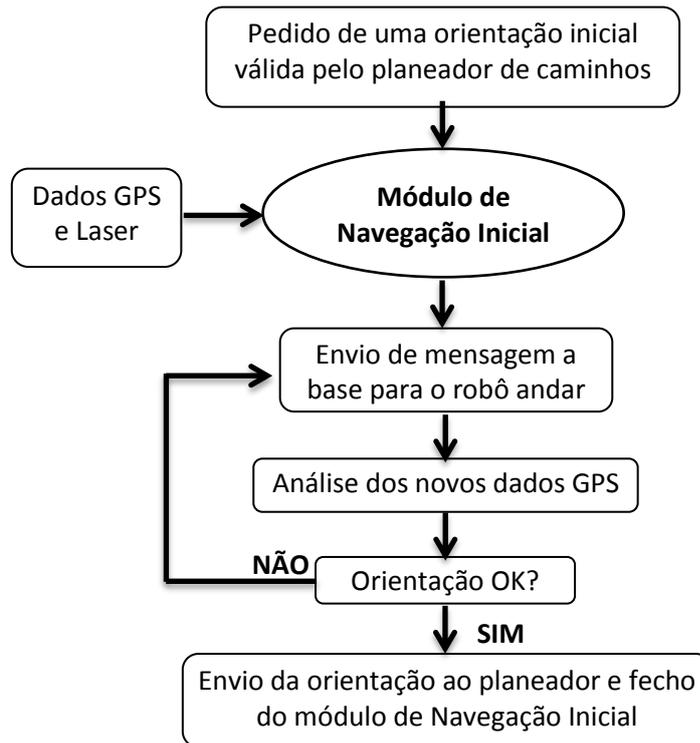


Figura 53 - Diagrama de funcionamento do módulo de Navegação Inicial

4.3.5. Planeamento de Caminhos

Inicialmente estava prevista a implementação de apenas um tipo de planeamento de caminhos por campos de potencial, mas durante o desenvolvimento deste trabalho surgiu um conceito novo de um planeamento por campos de potencial. O planeamento convencional relaciona todo o espaço em estudo com uma matriz cujos valores representam os valores do potencial da respetiva célula; o novo método implementado consiste na análise de apenas das zonas mais próximas em torno do robô, podendo ser considerado um planeamento local. As células de interesse encontram-se todas à mesma distância radial do robô, razão pela qual denominou-se o planeador de planeador radial. Um planeamento deste género possui mais vantagens, como por exemplo, o número de células ser bastante reduzido, diminuindo a necessidade de tempo de computação, e uma maior oferta de direções possíveis para o robô tomar no planeamento local, como se descreve mais à frente.

4.3.5.1. Planeamento de Trajetórias por Campos de Potencial Matriciais

Criou-se um módulo dedicado ao planeamento de trajetórias por campos de potencial matricial, ilustrado na Figura 54, onde todo o espaço em análise é discretizado numa matriz em que cada célula representa uma área do mapa e o cujo valor indica o potencial da respetiva zona. Este módulo



subscrive as mensagens do GPS, laser e orientação do robô, e publica uma mensagem contendo a direção e velocidade desejadas para chegar ao próximo destino. O programa inicia com a seleção da zona pelo utilizador, confirmando-o através de pontos GPS já conhecidos; depois de confirmado é carregado o respetivo mapa e é efetuada a sua discretização em células; estes processos encontram-se ilustrados na Figura 55. Esta divisão de área pode ser calibrada para qualquer dimensão desejada, sendo apenas necessário alterar os parâmetros da dimensão da célula desejada no algoritmo. A equação (10) representa a divisão horizontal do espaço, em que, X_{real} representa a dimensão real do mapa em x e Δx a distância real desejada para cada célula, obtendo-se assim, o número de células horizontais da matriz de comparação. A obtenção do número de células verticais é um processo idêntico que relaciona as distâncias em y .

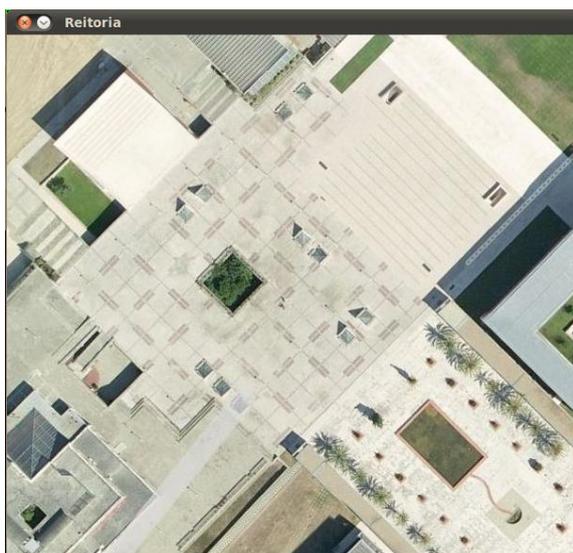


Figura 54 - Módulo de Planeamento de campos de potencial matricial

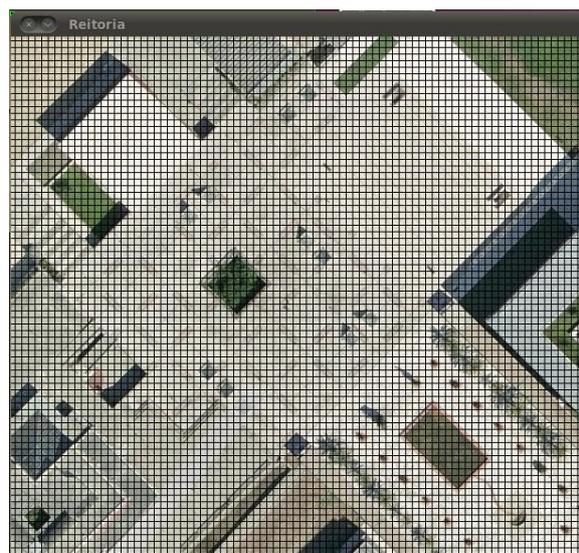
```
typedef struct {
    int dir;
    int orient_ok;      (validação da orientação)
    double gradiente;  (gradiente entre as células)
    int move;          (ordem para navegação inicial)
    int radial;        (distinção entre planea. matricial e radial)
    double timestamp;
    char *host;
}mppf_data_message;
```

Estrutura 13 - Mensagem do planeador

$$n_x = \frac{X_{real}}{\Delta x} \quad (10)$$



(a)



(b)

Figura 55 - Planeamento matricial - exemplo de carregamento e discretização do mapa
(a) – Carregamento do mapa (b) – Discretização em células



Os pontos de destino utilizados no algoritmo também são definidos antes do arranque do programa, sendo necessário definir as coordenadas geográficas dos mesmos. A passagem por estes pontos é realizada pela ordem pela qual são inseridos; outros pontos podem ser adicionados, ou os existentes podem facilmente ser alterados. Os obstáculos estáticos do mapa também podem ser pré colocados no programa; este passo é essencial para a limitação da zona de trabalho no mapa e para a definição das localizações de obstáculos negativos. Obstáculos negativos são obstáculos que se encontram a uma altura negativa em relação ao solo, como por exemplo buracos ou degraus. Atualmente o robô possui apenas um laser que se encontra direcionado na horizontal, tornando a deteção destes obstáculos impossível; seria necessário um segundo laser direcionado para o solo para conseguir detetar variações negativas da altura do solo. Todos os obstáculos negativos, os limites da zona e alguns obstáculos estáticos são previamente colocados no mapa de modo a ter uma representação mais correta do problema. Após a colocação destes obstáculos, são colocados os pontos destino pela ordem que serão passados; por sua vez, a posição atual do robô está constantemente a ser calculado e representada. Os obstáculos detetados pelo sensor laser não são representados no mapa, apenas interferem nos cálculos do campo de potencial global. Na Figura 56 encontra-se um exemplo da colocação dos vários pontos; os obstáculos estáticos são representados por pontos vermelhos, o ponto destino por um verde e a posição atual por um azul.

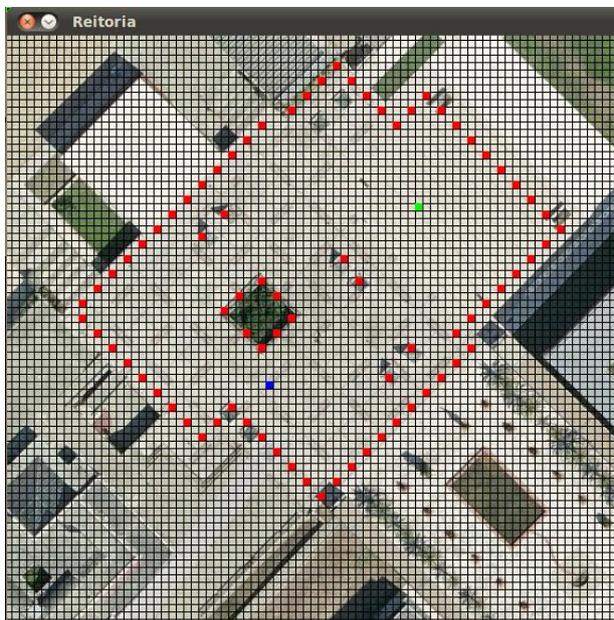


Figura 56 - Planeamento matricial - exemplo de colocação de obstáculos, destino, e posição atual

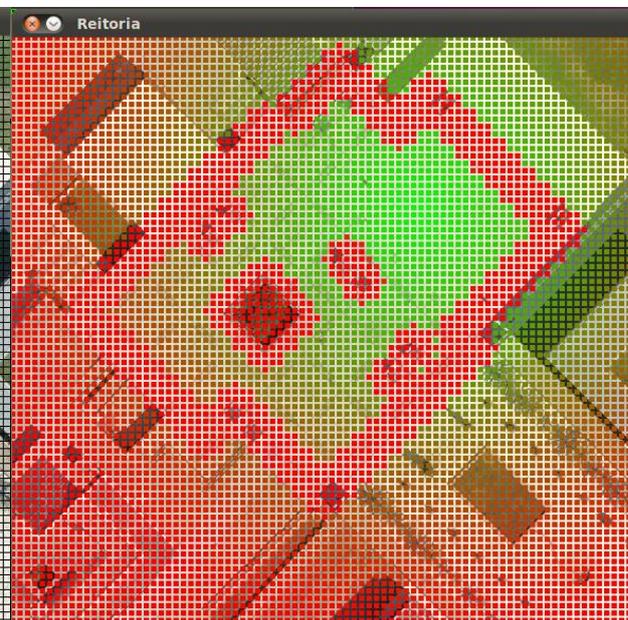


Figura 57 - Planeamento matricial - visualização do campo potencial

Antes da definição de todas estas características, o módulo de planeamento envia uma mensagem ao módulo de navegação inicial a pedir uma orientação inicial viável, que por sua vez executa o seu algoritmo de modo a obter esta orientação. Com todas as variáveis conhecidas,



localização e orientação, o programa executa uma série de funções de modo a calcular os campos de potencial atrator e repulsor, que são calculadas com as funções referidas no capítulo 2.3.3. Na Figura 57 encontra-se a visualização do campo potencial global criado para o exemplo em análise (Figura 56), onde se ilustram as células com maior valor de campo de potencial pela cor vermelha, significando uma repulsão do robô em relação a estas, com o decréscimo do campo de potencial as células tornam-se mais atrativas, ilustrado pelo verde. É bem visível o forte campo repulsor que os obstáculos estáticos geram, tornando as células próximas avermelhadas. As distâncias utilizadas nestas funções de cálculo são sempre as distâncias entre as células utilizadas na matriz, incluindo os obstáculos captados pelo sensor laser que são colocados na célula correspondente de modo a usar o mesmo referencial. A mudança de referencial, do referencial local do robô para o referencial global da matriz, dos obstáculos detetados pelo laser está dependente da orientação do robô, o *true_course*. Na Figura 58 encontram-se ilustrados os dois referenciais, onde se verificam as diferenças entre eles e a dependência da rotação do eixo local na orientação do robô. A mudança de referencial é conseguida através da aplicação das equações (11), (12), (13) e (14), em que as primeiras duas descobrem as distâncias, em células, dos obstáculos em relação ao robô no referencial global e as últimas duas colocam os obstáculos nas células correspondentes no referencial global. A informação laser recebida contém a distância entre o robô e os vários obstáculos detetados, em metros e no referencial local. Assim, é aplicada uma mudança de referencial mencionada, de rotação, de modo a obter as distâncias nos eixos do referencial global, e uma divisão pela distância real de cada célula ($\Delta real_x$ e $\Delta real_y$) de modo a obter a distância a cada obstáculo em células. Após a obtenção das distâncias dos obstáculos laser em células e no referencial global, com aplicação das equações (11) e (12), são somadas à célula correspondente à posição atual do robô para obter a célula correspondente de cada obstáculo na matriz global, como demonstrado nas equações (13) e (14). Este processo é realizado para todos os obstáculos detetados pelo laser e sempre que é detetado um novo obstáculo.

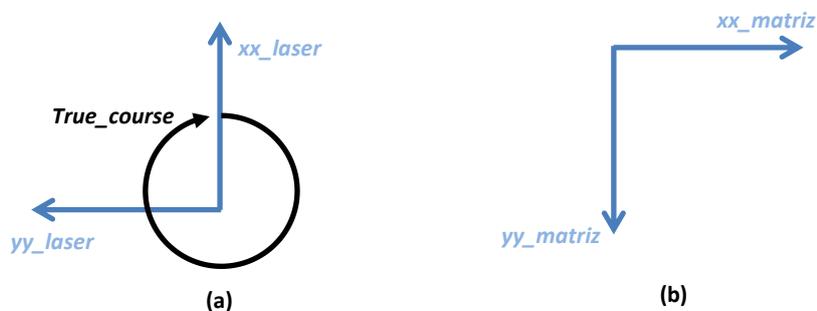


Figura 58 - Comparação entre referenciais
(a) – Referencial local (b) – Referencial global



$$xx_{obs.global} = \frac{xx_{obs.laser} * \sin(true_course)}{\Delta real_x} - \frac{yy_{obs.laser} * \cos(true_course)}{\Delta real_y} \quad (11)$$

$$yy_{obs.global} = -\frac{xx_{obs.laser} * \cos(true_course)}{\Delta real_x} - \frac{yy_{obs.laser} * \sin(true_course)}{\Delta real_y} \quad (12)$$

$$xx_{obs.global} = xx_{obs.global} + xx_{current} \quad (13)$$

$$yy_{obs.global} = yy_{obs.global} + yy_{current} \quad (14)$$

Posteriormente à colocação dos obstáculos nas células correspondentes, na matriz global, realiza-se o cálculo do potencial repulsor de cada um destes em relação ao robô. Mesmo que uma célula contenha vários obstáculos apenas é contabilizada uma vez no cálculo, não existe distinção entre células com um ou vários obstáculos, basta uma célula conter um obstáculo para ser considerada uma célula repulsora. Após a obtenção do campo de potencial global, cada célula do mapa é associada ao seu respectivo valor calculado. O cálculo está constantemente a ser efetuado de modo a que seja influenciado por qualquer alteração que possa haver do meio. Consequentemente, cada vez que é calculado o campo de potencial é realizada a análise das células vizinhas à célula correspondente à posição atual do robô. O objetivo desta análise consiste na obtenção da célula vizinha com menor valor de campo de potencial, que será a próxima célula destino local. Este ciclo de cálculo encontra-se ilustrado na Figura 59.

Sabendo a localização da célula com menor potencial, na matriz local, o tamanho real das células e a orientação real do robô, consegue-se calcular a direção necessária para o robô atingir a célula destino. O primeiro passo neste cálculo é a obtenção da localização da próxima célula destino e da orientação real do robô, sendo conseguido através do índice da célula escolhida, representados na Figura 60 e na Tabela 5, e a pela informação obtida pelo recetor GPS. A zona de análise é constituída pelas oito células exteriores de uma matriz de 3x3, tornando-as os destinos locais possíveis no planeamento do caminho. O segundo passo consiste na aplicação de pequenas equações, exprimidas na Equação (15), (16) e (17), de modo a obter a direção ótima para chegar a célula seguinte. A equação principal da Equação (15), obtém a diferença angular entre a orientação do robô a localização do próximo destino local, as equações secundárias garantem que o resultado seja sempre o ângulo mais pequeno entre as duas células. Considerou-se os resultados positivos para as direções orientadas para a direita e negativos para a esquerda.

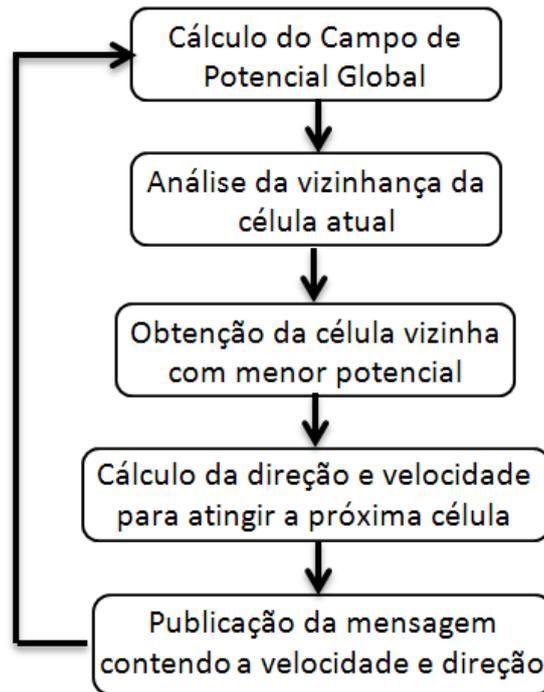


Figura 59- Planejamento Matricial - Sequência de ações

$$\theta = orient_{prox.celula} - orient_{robo} \quad (15)$$

$$\begin{aligned} &if (\theta < -180) \\ &(\theta = \theta + 360) \end{aligned} \quad (15.a)$$

$$\begin{aligned} &if (\theta > 180) \\ &(\theta = \theta - 360) \end{aligned} \quad (15.b)$$

A direção necessária para o robô atingir o próximo destino local está dependente do tamanho real das células, visto que é assumido que tanto o robô como o destino local se encontram no centro da sua respectiva célula. Também se assumiu que os comprimentos das células são iguais nas duas direções (dx_{real} e dy_{real}). O passo seguinte será a obtenção do raio do arco da trajetória que une a célula atual com a célula destino, que é conseguida através da equação (16) [20], onde L representa a distância e θ o ângulo entre os dois pontos, que foi anteriormente calculado. Os valores de L estão representados na Tabela 5, em que d representa o valor da distância real das células ($delta_x_real$). Por razões de cálculo definiu-se o valor de aproximadamente zero para a orientação da célula número dois. Com o valor do raio do arco conhecido, Rb , é possível calcular o ângulo de viragem das rodas para o robô atingir o ponto destino através da equação (17) [20]. Na Figura 61 encontra-se esquematizado a relação entre arco de uma curva e o ângulo de viragem do robô.



Tabela 5 - Planeamento Matricial – Orientações e distância das células destino

Nº da célula destino	$orient_{prox.celula}$ - (º)	Distância entre centros – L (m)
1	325	$\sqrt{2}d$
2	≈ 0	d
3	45	$\sqrt{2}d$
4	270	d
5	90	$\sqrt{2}d$
6	225	d
7	180	$\sqrt{2}d$
8	135	d

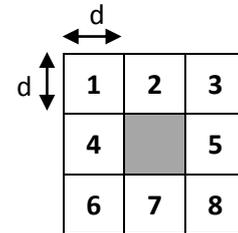


Figura 60- Numeração das células vizinhas

$$Rb = \frac{L}{2 \sin\left(\frac{\theta}{2}\right)} \quad (16)$$

$$\varphi = \tan^{-1}\left(\frac{1}{Rb}\right) \quad (17)$$

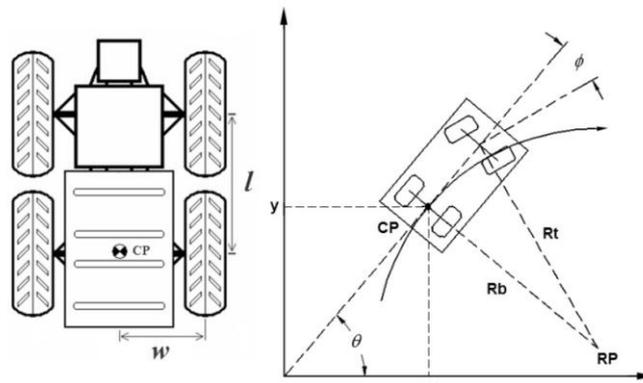


Figura 61 - Planeamento Matricial, obtenção da direção [20]

Na Figura 62 encontra-se ilustrado um exemplo genérico deste processo, com células de dois por dois metros, como robô na célula central orientado a 90º (desde o Norte geográfico e no sentido dos ponteiros dos relógios). Presume-se que a próxima célula destino se localiza a 45º, como representado na figura.

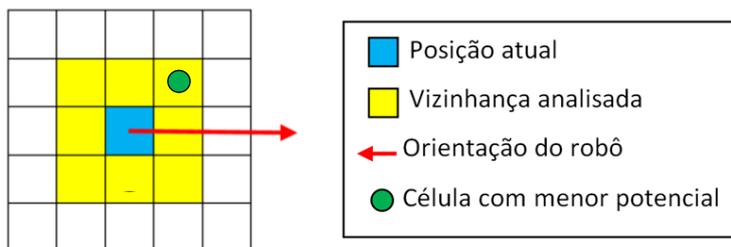


Figura 62 - Planejamento Matricial - mudança de referencial (exemplo)

Aplicando a equação (15) obtém-se:

$$\theta = 45 - 90 = -45^\circ$$

Aplicando a equação (16) e (17) obtém-se:

$$Rb = \frac{\sqrt{2} * 2}{2 \sin\left(\frac{-45}{2}\right)} = -1.848 \text{ m}$$

$$\varphi = \tan^{-1}\left(\frac{1}{-1.848}\right) \approx -28.42^\circ$$

Assim conclui-se que o robô necessita de virar 28.42° , para a esquerda, para alcançar a célula desejada. Esta informação seria encaminhada para o módulo base do robô que comunica com o *hardware* específico para realizar a tarefa. O sistema atual do robô permite um ângulo de viragem máximo de 23° , para ambos os lados, caso o valor pedido seja superior é este que é utilizado.

Esta análise e escolha de célula é realizada de maneira cíclica até o robô atingir o ponto objetivo global; devido ao erro de leitura do GPS, considerou-se uma tolerância entre a localização do robô e o ponto destino, não sendo necessário atingir as coordenadas exatas, apenas tem de se encontrar “perto” deste. A distância de tolerância também é um parâmetro que pode ser facilmente alterado no programa. Uma das principais desvantagens do planejamento por campos de potencial matricial é a oferta de destino local, visto que a vizinhança de análise é constituída apenas por oito células limitando os destinos locais a estas oito.

Todo o programa foi realizado de modo a poder alterar qualquer variável que intervém no cálculo dos potenciais, desde o tamanho das células até as constantes atratora e repulsora. A mensagem publicada contém informação da direção e velocidade, a serem interpretadas pelo módulo base, o requerimento de uma orientação inicial válida, interpretada pelo módulo de



navegação inicial, e ainda uma variável para distinguir entre os dois tipos de planeadores de campos de potencial, visto que ambos os planeadores publicam a mesma mensagem, detalhada na Estrutura 13.

4.3.5.2. Planeamento de Trajetórias por Campos de Potencial Radial

Durante a realização deste trabalho surgia a ideia de implementar um método de planeamento novo, que em vez de calcular o campo de potencial de todo o espaço, apenas é calculado para algumas células em torno do robô. Um planeador deste género permite definir o número de pontos de interesse em torno do robô, possibilitando um maior número de pontos da vizinhança analisada, tornando possível uma maior oferta de direções para o robô tomar. Decidiu-se implementar um método em que os pontos de interesse estivessem todos localizados à mesma distância do robô, a uma distância radial R , como ilustrado na Figura 63. Devido a esta característica, o módulo dedicado a este planeador foi denominado de planeamento por campos de potencial radial. Este módulo possui as mesmas características de comunicação e mensagens do módulo de planeamento matricial, pormenorizadas na Figura 54 e Estrutura 13.

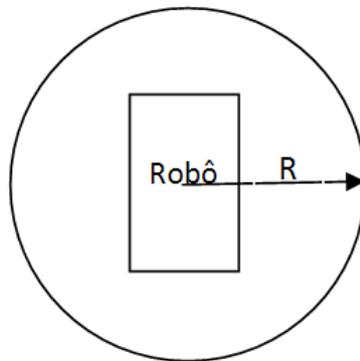


Figura 63 - Planeamento Radial - Zona de interesse

Como este planeador não associa nenhum elemento comparativo com o mapa da zona em análise, optou-se por usar as distâncias reais entre os pontos de interesse: pontos obstáculos, posições atuais e pontos destinos. Decidiu-se usar a fórmula *Haversine* [19] por este possuir a capacidade de calcular distâncias reais entre dois pontos através das suas coordenadas GPS, este cálculo constitui-se por três equações, a equação (18), (19) e (20), que têm como entrada os valores de latitude e longitude dos pontos em interesse. Além da distância entre dois pontos também será necessário saber a posição de um ponto em relação ao outro, ou seja, a orientação da linha que os



une. O cálculo desta orientação é conseguido através da equação (21), que consegue calcular a posição angular de uma coordenada geográfica em relação a outra. Esta orientação é calculada no mesmo referencial da orientação encaminhada pelo recetor GPS, ou seja, com o zero alinhado com o Norte geográfico e com um comportamento crescente no sentido dos ponteiros do relógio. Com as localizações, relativas entre pontos, conhecidas é possível estabelecer as relações entre o ponto atual do robô e os pontos destino e de obstáculos estáticos conhecidos.

$$a = \sin^2\left(\frac{\Delta lat}{2}\right) + \cos(lat_1) * \cos(lat_2) * \sin^2\left(\frac{\Delta long}{2}\right) \quad (18)$$

$$c = 2 * \text{atan2}(\sqrt{a}, \sqrt{1-a}) \quad (19)$$

$$d = R * c \quad (20)$$

(R – Raio de Terra = 6371km)

$$\theta = \text{atan2}(\sin(\Delta lat) \cos(lat_2), \cos(lat_1) \sin(lat_2) - \sin(lat_1) \cos(lat_2) \sin(\Delta lat)) \quad (21)$$

Ao contrário do planeador anterior, todos os cálculos de potencial atrator e repulsor são realizados em relação ao referencial local, nos pontos de interesse em torno do robô. Através das funções descritas anteriormente (equações (18), (19), (20) e (21)) consegue-se saber as posições relativas entre os pontos de interesse do robô e os pontos de destino e de obstáculos estáticos conhecidos. Os obstáculos detetados pelo sensor laser não sofrem nenhuma transformação visto que já se encontram no referencial do robô, por razões de poupança de cálculo apenas são considerados os obstáculos pontuais detetados pelo laser que estejam espaçados a 20 centímetros entre si. Reunidas estas condições, o algoritmo está habilitado a calcular o campo de potencial.

Decidiu-se usar onze pontos de interesse na parte dianteira do robô, oferecendo deste modo cinco direções para cada lado e uma direção nula central para o planeador escolher durante a sua execução. Na Figura 64 encontra-se a zona de interesse com as células pontuais representadas por círculos verdes; o ponto azul é apenas de referência para saber a orientação com que o robô se encontra em relação ao referencial global. O número de pontos de interesse pode facilmente ser alterado, não sendo necessário fazer uma nova calibração da direção, visto que esta se encontra em função do ângulo do ponto de interesse selecionado.

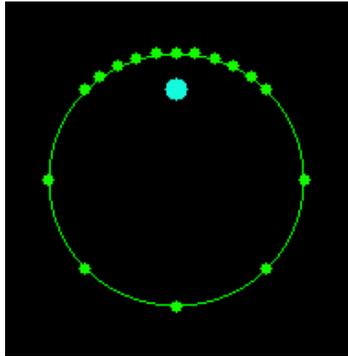


Figura 64 - Planeamento Radial - Pontos de interesse

Este planeador possui a mesma sequência de ações do planeador anterior, iniciando com a escolha da zona, seguido da colocação dos obstáculos, pontos destino e ponto atual, e pela obtenção da orientação válida através de um pedido ao módulo de navegação inicial. Seguido desta sequência, o módulo entra no mesmo ciclo de cálculo do planeador anterior, representado na Figura 59. As únicas diferenças encontram-se no número de pontos analisados e na forma como são calculados os campos de potencial; neste caso são utilizadas valores reais das distâncias (em metros). Neste planeamento apenas são calculados os potenciais dos pontos da zona de interesse à volta do robô. Com a análise destes pontos descobre-se a célula pontual com menor potencial de modo a obter a direção necessária para chegar à respetiva célula. O cálculo desta direção segue a mesma filosofia do planeador matricial, contendo apenas diferenças nos valores atribuídos ao ângulo e à distância entre a célula atual e a destino. A distância entre as células é sempre a mesma, sendo o valor do raio (R) da zona de interesse atribuído no início do programa, e o valor da distância angular que separa as duas células, é a mesma que foi atribuída na separação das células radiais. Os valores das distâncias angulares para o modelo proposto, 16 pontos de interesse em torno do robô, encontram-se listados na Tabela 6. Considerou-se a célula de índice número um a que se encontra na parte dianteira do robô, aumentando o índice no sentido dos ponteiros do relógio. Os cálculos seguintes são iguais ao planeador anterior, sendo necessário aplicar a equação (15), (16) e (17) para descobrir o ângulo da direção a enviar ao módulo base para atingir a célula destino.

Na Figura 65 encontra-se um exemplo de execução deste módulo, onde no canto inferior direito são representados os pontos de interesse, pelos pontos verdes, e o ponto com menor potencial escolhido, pelo ponto vermelho. Estas representações estão em atualizações constantes de modo a acompanhar a alteração da orientação do robô.



Tabela 6 - Planeamento Radial – Orientações das células destino

Nº da célula destino	$orient_{prox.celula}$ - (º)	Nº da célula destino	$orient_{prox.celula}$ - (º)
1	≈0	9	180
2	9	10	225
3	18	11	270
4	27	12	315
5	36	13	324
6	45	14	333
7	90	15	342
8	135	16	351

O funcionamento base deste planeador é semelhante ao matricial, sendo baseada no cálculo do campo de potencial e na análise dos pontos da área de interesse de modo a determinar o ponto com o potencial mais baixo. Com esta determinação e sabendo a orientação verdadeira do robô, calcula-se a direção e velocidade necessária para o robô chegar ao ponto objetivo. Os valores da direção e velocidade desejadas são publicados na mensagem do módulo de modo que possa ser interpretada pelo módulo base do robô.

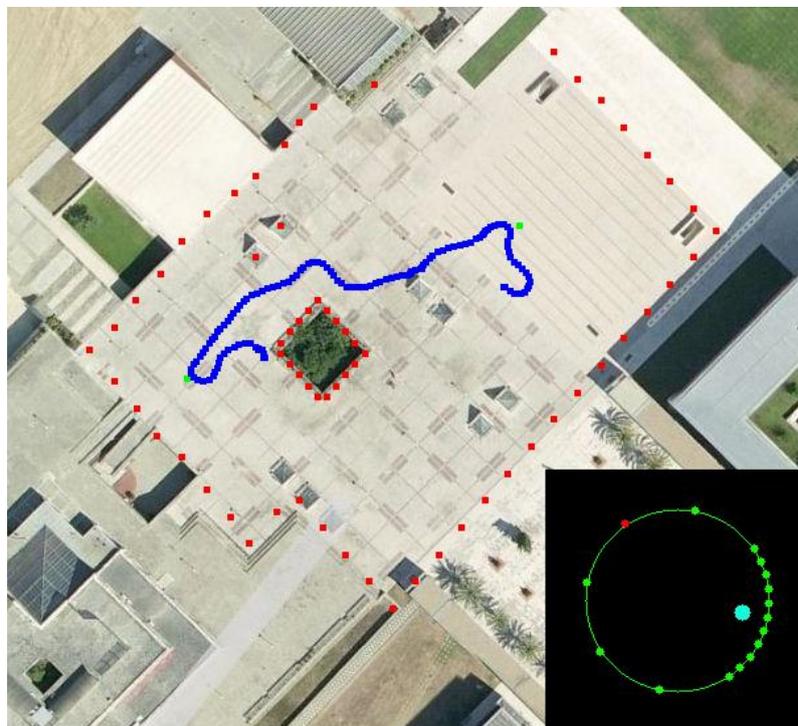


Figura 65 - Planeamento Radial - exemplo



4.3.6. Funcionamento Global

Até ao momento foram descritos os módulos criados, o funcionamento de cada um e descritas as trocas de mensagens que efetuam; neste subcapítulo será enumerado a sequência de funcionamento que sincroniza todos estes módulos. Para facilitar o arranque dos módulos necessários para a navegação por GPS criou-se um *Process Control* dedicado a este efeito, ilustrado na Figura 66. Acrescentou-se o módulo de *Gamepad* à esta lista por ser muito utilizado em *debug*.



Figura 66 - *Process Control* da Navegação por GPS

O arranque do sistema inicia-se sempre com a ativação do módulo Central, que é fundamental para efetuar as trocas de mensagens entre os módulos; o segundo módulo a ser ativo é o *ParamDaemon2000* que obtém e define todas as variáveis que possam ser utilizadas pelos restantes módulos. Naturalmente, de seguida ativam-se os módulos responsáveis pela comunicação com o *hardware* do robô, o módulo *Base2000_1*, responsável pelos motores, o módulo *Laser*, encarregue de comunicar com o sensor laser, e o módulo *GPS*, que comunica com o sensor GPS montado no robô. A ordem de ativação destes últimos três módulos não é importante, visto que não comunicam entre si, mas todos os restantes têm de ser ativados pela ordem indicada. Com o estabelecimento da comunicação com o *hardware* realizada ativam-se os módulos de *Orientation*, *Ini_Navig* e *MPPF* (ou *MPPF_RADIAL*) pela ordem mencionada. Decidiu-se abreviar os nomes módulos de planeamento para *MPPF*, *Motion Planner with Potential Fields*. Ambos os módulos de planeamento foram adicionados à lista, sendo necessário escolher apenas um, dependendo do tipo de planeamento desejado, por campos de potencial matricial ou radial.

Com os módulos ativos pela ordem referida, o módulo de planeamento encaminha uma mensagem para o módulo de Navegação Inicial de modo a obter uma orientação inicial válida, o módulo de Navegação Inicial recebe mensagens do laser e autoriza, ou não, o robô a andar em reta



até obter a orientação viável. Uma vez obtida uma orientação válida, o valor desta é encaminhada para o módulo de planeamento e é enviada a ordem para o robô se imobilizar. Assim, o módulo de Navegação Inicial cessa as suas trocas de mensagens e é terminado. Com a orientação inicial obtida, o planeador de caminhos continua com o seu programa de modo a calcular as direções e velocidades necessárias para o robô atingir o ponto destino. Este módulo está em comunicação constante com os módulos Laser, GPS, Orientação e Base do robô. O funcionamento global do sistema encontra-se esquematizado na Figura 67

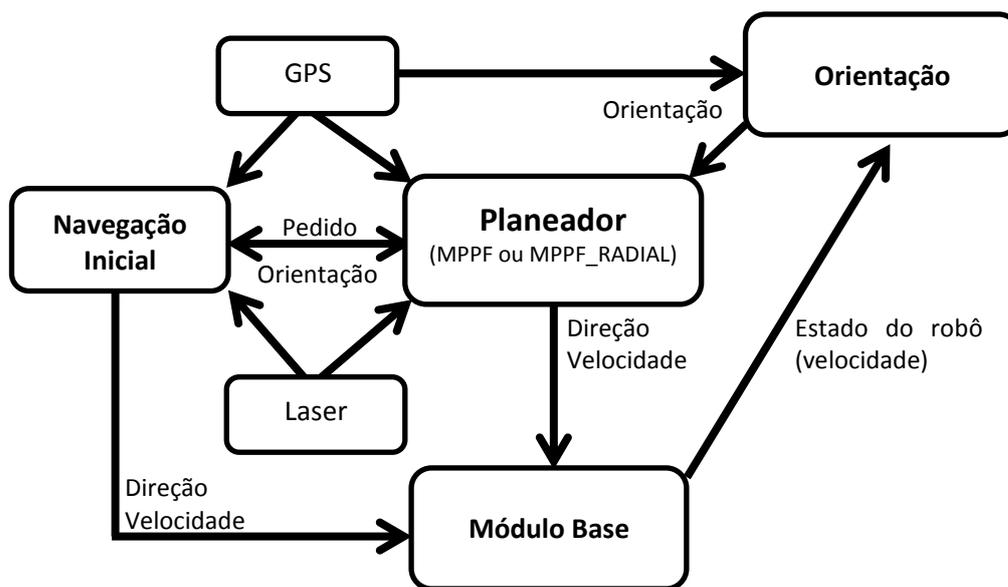


Figura 67 - Esquema do funcionamento global do planeamento por GPS



5. Ensaio e Resultados

Neste capítulo serão demonstrados alguns dos ensaios realizados dos sistemas desenvolvidos ao longo deste projeto e os principais resultados obtidos. Realizaram-se ensaios ao sistema de monitorização das baterias de modo a avaliar o seu desempenho e precisão, e também se realizaram vários ensaios aos dois sistemas de planeamento de caminhos por campos de potencial, o matricial e o radial.

5.1. Monitorização das baterias

De modo a obter a eficácia e precisão do sistema de monitorização de baterias, realizaram-se várias leituras a baterias através do sistema concebido e posteriormente através de um multímetro, de modo a obter o valor real das tensões. Ao todo, efetuaram-se cinco leituras de conjuntos de baterias com valores de carga diferentes, na Tabela 7 encontram-se os dados registados e também os erros associados à leitura das mesmas. Como é possível observar na tabela, os erros estão diretamente associados à posição a que se encontram na série, aumentando com o índice da bateria em análise, fenómeno que ocorre devido à sua ligação em série. Uma ligação deste género impossibilita a medição direta a cada bateria de modo independente, e permite apenas a leitura da carga acumulada em série. Ou seja, o valor de carga lido numa bateria está dependente das anteriores na série, apenas a primeira é que se encontra independente das restantes. A leitura de cada bateria está dependente da leitura das anteriores, tornando o erro associado a cada leitura acumulável. Assim é previsto que a bateria que se encontra no final da série, a número 4, contenha o maior erro de leitura. Analisando a Tabela 7 verifica-se que o erro médio das leituras das primeiras três baterias da série é entre 0,17 e 0,19 Volt, e na última este valor sobe até aos 0,3 Volt. O maior erro registado nos ensaios ocorridos foi de 0,64 Volt.



Tabela 7 - Monitorização das Baterias - ensaios realizados

Ensaio	Tipo de valor lido	Bateria 1	Bateria 2	Bateria 3	Bateria 4
1	Valor real (V)	12,9	13,01	12,79	13,07
	Valor lido (V)	13,06	13,19	12,94	13,13
2	Valor real (V)	12,88	12,11	12,93	12,94
	Valor lido (V)	13,06	12,26	13,12	13,11
3	Valor real (V)	12,81	12,97	12,83	12,15
	Valor lido (V)	13,06	13,00	12,94	12,79
4	Valor real (V)	12,51	12,61	12,94	12,78
	Valor lido (V)	12,69	12,91	13,25	12,45
	Erro médio (V)	0,19	0,17	0,19	0,30
	Erro max. (V)	0,25	0,30	0,31	0,64
	Erro min (V)	0,16	0,03	0,11	0,06
	Erro médio (%)	1,51	1,31	1,47	2,41
	Erro max. (%)	1,95	2,38	2,40	5,27
	Erro min. (%)	1,24	0,23	0,86	0,46

5.2. Planeamento por Campos de Potencial Matricial

Decidiu-se realizar ensaios e testes para retirar as principais conclusões deste trabalho, começando por realizar ensaios simples sem obstáculos e aumentando o nível de exigência com cada tipo de ensaio. Todos estes ensaios foram realizados com uma distância de tolerância entre o robô e ponto destino, e definiu-se o valor de duas células para esta tolerância. Durante os primeiros ensaios realizados registou-se que o aumento da velocidade proporciona um aumento do erro de localização e orientação do GPS. Decidiu-se assim, usar uma velocidade constante de valor reduzida de modo a diminuir erros de posicionamento, obteve-se experimentalmente o valor de 0,8 m/s. O uso de uma velocidade constante em todos os ensaios também permite uma melhor comparação entre os diversos resultados obtidos, sujeitando todos os ensaios à mesma restrição de velocidade. O sensor laser utilizado possui um alcance de 5,6 metros, assim decidiu-se nunca usar um valor de distância de influência dos obstáculos inferior a este.

- **Ensaio do Tipo 1:**

O primeiro ensaio realizado consistiu na passagem por dois pontos de destino relativamente próximos, de modo alternado, e sem obstáculos entre eles. Quando o robô atinge um ponto destino segue para o outro, criando assim um circuito interminável entre os dois pontos; na Figura 68



encontra-se a localização e ordem de passagem destes pontos. Realizaram-se vários ensaios deste tipo alterando os parâmetros que mais influenciam o planeamento de trajetórias, a constante atratora e a dimensão das células. A distância de influência dos obstáculos e a constante repulsora não foram alterados visto que este ensaio tinha como objetivo avaliar o desempenho do campo atrator gerado pelos pontos destino. Na Tabela 8 encontram-se listados os parâmetros usados em três dos ensaios realizados, e os seus resultados se encontram na Figura 69, Figura 70 e Figura 71.

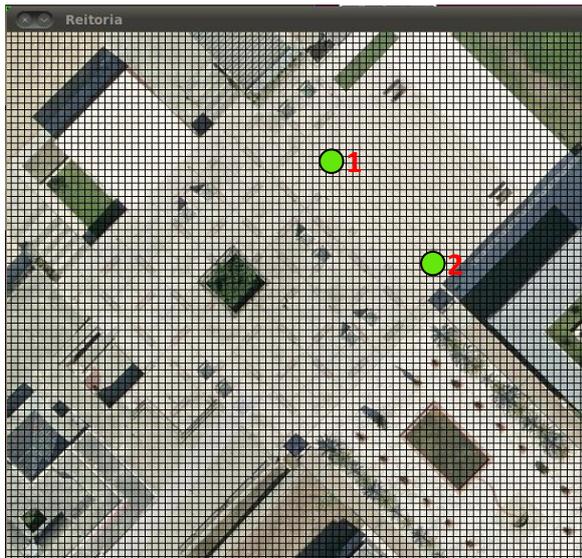
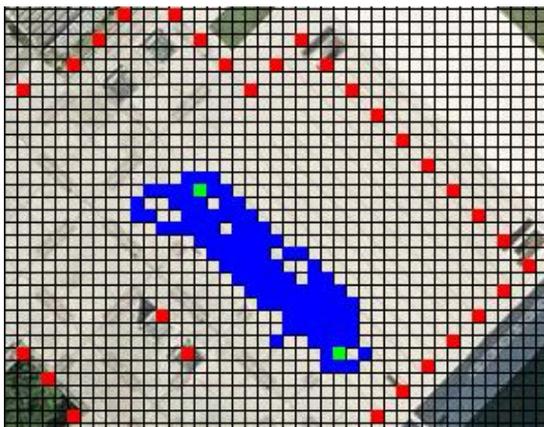


Figura 68 - Planeador Matricial - Circuito do ensaio do tipo 1

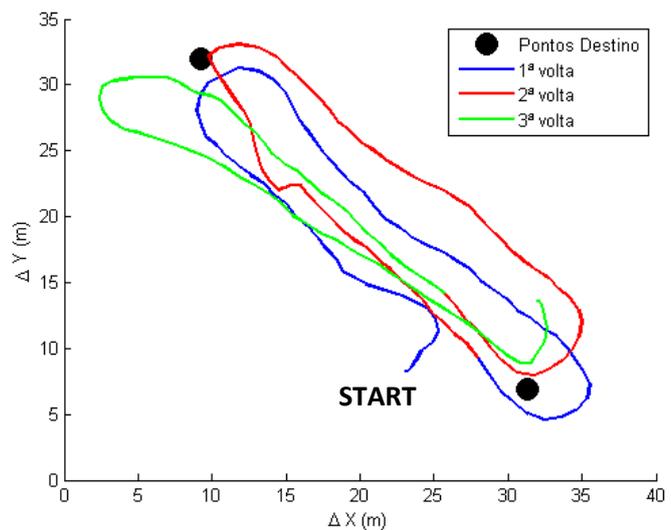
Tabela 8 - Planeamento Matricial - Ensaios realizados do Tipo 1

Ensaio	1	2	3
E (constante atrator)	1	15	15
N (constante repulsor)	10	10	10
Distância de influência (células)	4	4	4
Dimensão da célula (metros)	2x2	2x2	4x4

• **Resultados do ensaio do tipo 1:**



(a)

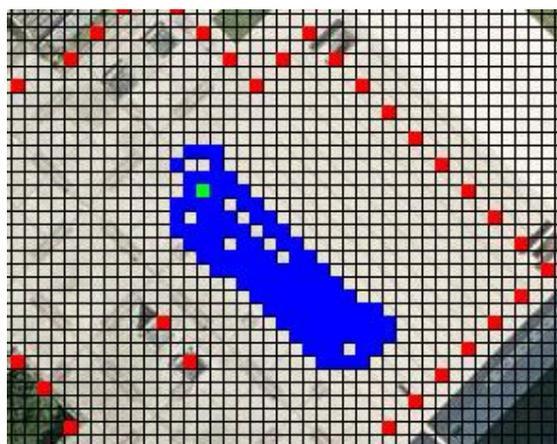


(b)

Figura 69 - Planeamento matricial - ensaio do tipo 1 – 1

(a) – Representação no mapa (parte)

(b) - Representação das voltas realizada



(a)

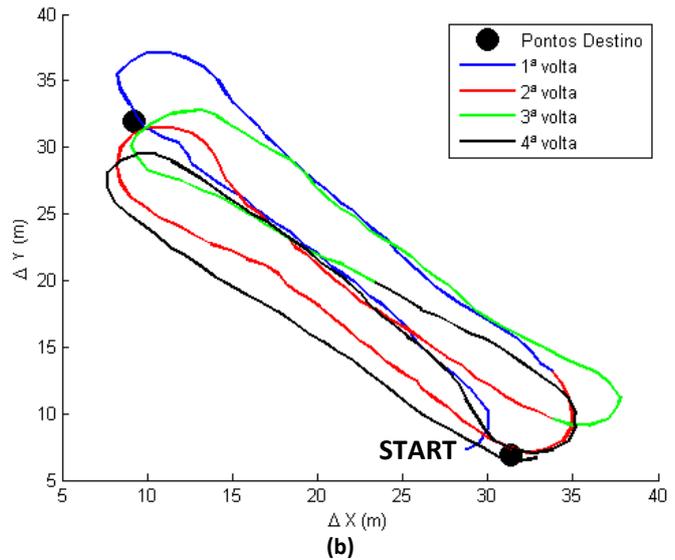
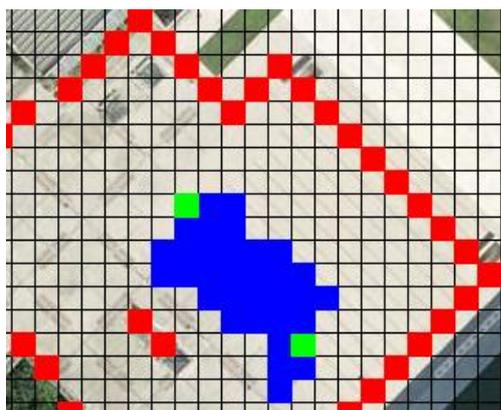


Figura 70 - Planejamento matricial - ensaio do tipo 1 - 2

(a) - Representação no mapa (parte)

(b)- Representação das voltas realizadas



(a)

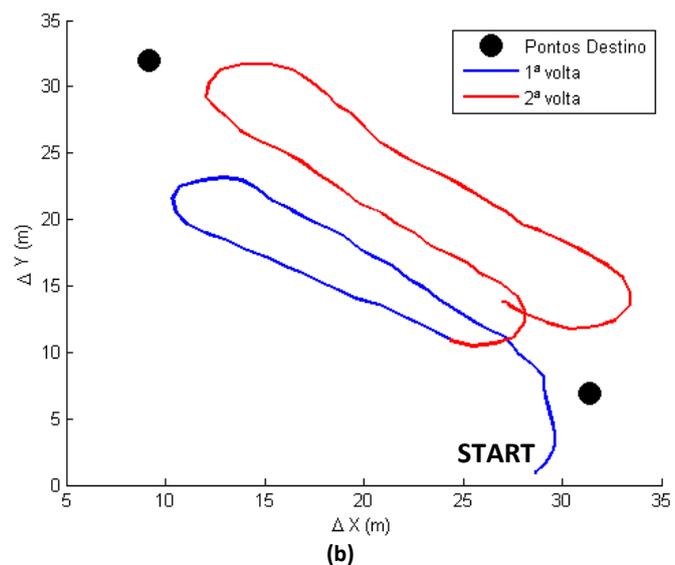


Figura 71 - Planejamento matricial - ensaio do tipo 1 - 3

(a) - Representação no mapa (parte)

(b)- Representação das voltas realizadas

Com a realização do ensaio do tipo 1 observou-se que, com o aumento da constante atratora o robô é atraído com mais “força” para o destino, percorrendo caminhos mais regulares. Este fenómeno é observável comparando o ensaio 1-1 (Figura 69) com o ensaio 1-2 (Figura 70), onde se verifica caminhos mais regulares no segundo ensaio. No ensaio 1-3 (Figura 71) utilizaram-se células maiores, de 4 metros de lado, para analisar a sua influência no planeamento de caminhos, e enquanto não se verificam grandes alterações nos caminhos realizados, constata-se um maior erro de posicionamento. Observando estes resultados, conclui-se que o cálculo do campo atrator está a ser bem concretizado e que células de 4 metros são desfavoráveis no planeamento, pois influenciam



negativamente os resultados. Em todos os ensaios realizados é possível observar a fase inicial de obtenção de uma orientação válida, momento que o robô se desloca em linha reta até obter uma orientação válida.

• **Ensaio do tipo 2**

Para o segundo tipo de ensaio decidiu-se inserir mais um ponto no circuito anterior, criando um circuito triangular, os ensaios ocorridos neste circuito são semelhantes aos anteriores, servem meramente para avaliar o comportamento do robô em meios sem obstáculos. O circuito utilizado encontra-se ilustrado na Figura 72 e alguns dos ensaios realizados encontram-se listados na Tabela 9.

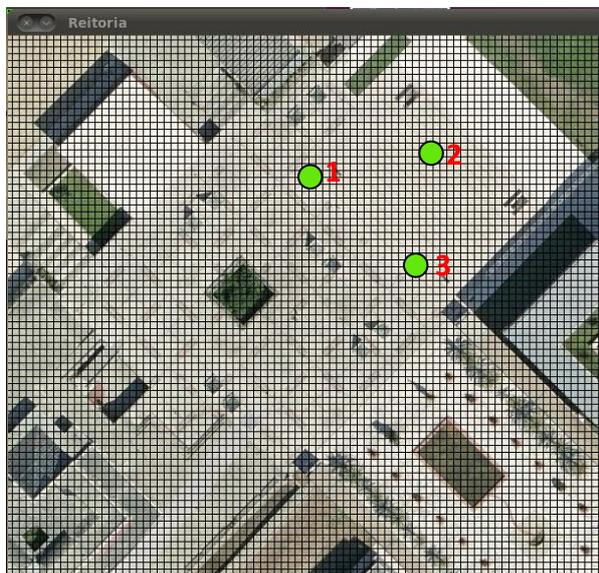
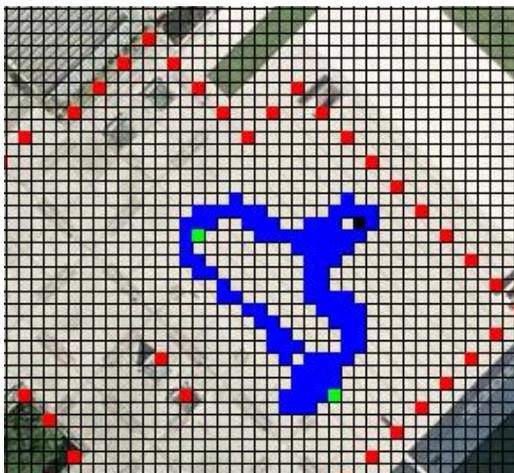


Figura 72 - Planeador Matricial - Circuito do ensaio do tipo 2

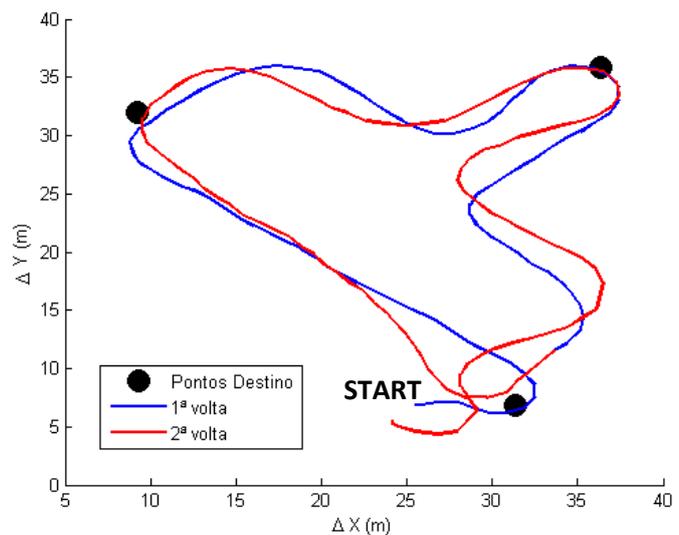
Tabela 9 - Planeamento Matricial - Ensaio realizados do Tipo 2

Ensaio	1	2	3
E (constante atrator)	1	15	10
N (constante repulsor)	10	10	10
Distância de influência (células)	4	4	4
Dimensão da célula (metros)	2x2	2x2	4x4

• **Resultados dos ensaios do tipo 2:**



(a)

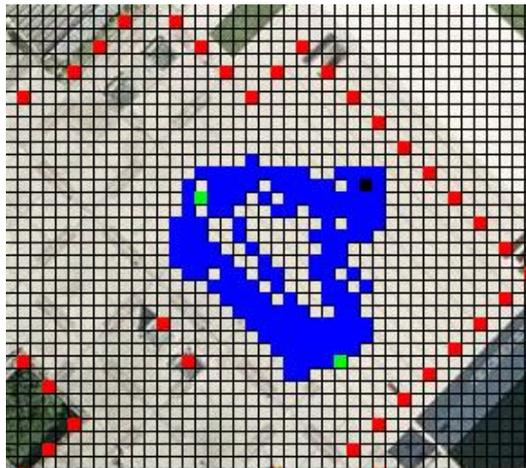


(b)

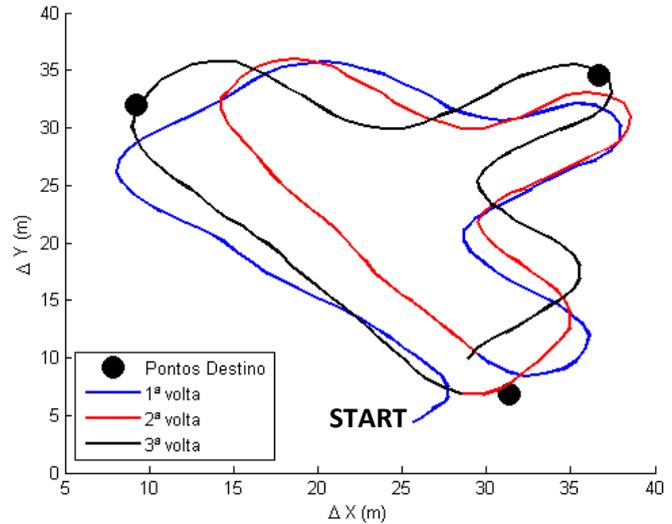
Figura 73 - Planeamento matricial - ensaio do tipo 2 – 1

(a) Representação no mapa (parte)

(b)- Representação das voltas realizadas



(a)

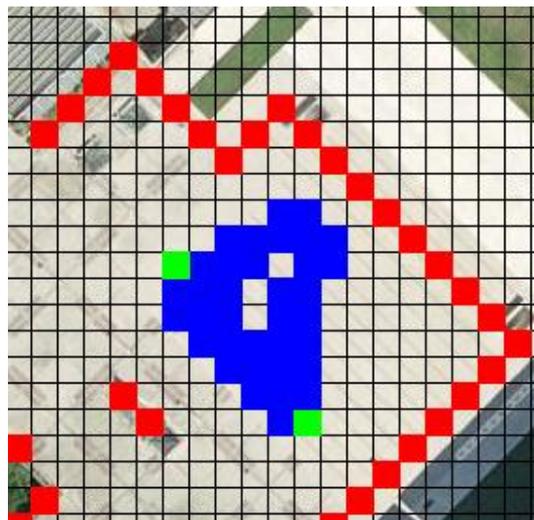


(b)

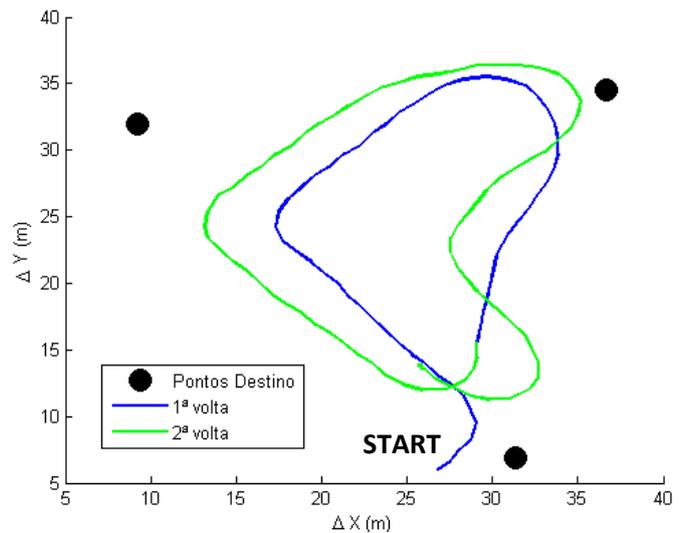
Figura 74 - Planejamento matricial - ensaio do tipo 2 – 2

(a) – Representação no mapa (parte)

(b) - Representação das voltas realizadas



(a)



(b)

Figura 75 - Planejamento matricial - ensaio do tipo 2 – 3

(a) – Representação no mapa (parte)

(b) - Representação das voltas realizadas

Nestes ensaios verificou-se pouca influência da constante atratora na realização dos caminhos; ao comparar os ensaios 2-1 (Figura 73) e 2-2 (Figura 74) verifica-se que, com o aumento da constante atratora os caminhos tornam-se apenas ligeiramente mais regulares. Estes ensaios demonstram alguns caminhos serpenteados; o seu aparecimento era esperado, como mencionado anteriormente, e devem-se a vários fatores, entre as quais, o atraso da atualização da orientação enviada pelo GPS, o erro de posicionamento do GPS, e também pela limitação de destinos locais possíveis na vizinhança analisada em cada célula (oito destinos possíveis). Realizou-se mais um



ensaio deste tipo mas com células de maior dimensão, ensaio 2-3 (Figura 75), em que até se observa caminhos mais regulares mas com maior erro de posicionamento. Os caminhos mais regulares devem-se ao aumento do tamanho das células; células de maior dimensão permitem que o robô percorra uma distância maior até “descobrir” que se encontra na célula seguinte.

- **Ensaio do tipo 3**

Este tipo de ensaio tem como principal objetivo avaliar o comportamento do robô com a inclusão de obstáculos virtuais no seu caminho. Obstáculos virtuais são obstáculos conhecidos e estáticos inseridos manualmente no programa antes do arranque do mesmo, têm o mesmo peso no cálculo do potencial repulsor que os obstáculos reais detetados pelo laser. Decidiu-se separar os ensaios com obstáculos virtuais dos ensaios com obstáculos reais para comparar os resultados destes, visto que os obstáculos virtuais influenciam sempre a escolha da trajetória, desde que se encontram à distância de influência. E por sua vez, os obstáculos reais apenas entram na escolha de trajetória quando são detetados pelo laser, e se encontram à distância de influência, podendo ser considerados obstáculos temporários. A partir do momento que não são detetados pelo laser, deixam de influenciar o cálculo de trajetória, mesmo que ainda se encontram à distância de influência. Colocou-se um obstáculo virtual entre dois pontos de destino e realizaram-se ensaios com parâmetros diferentes de modo a verificar a influência de cada um destes na escolha de trajetórias; o circuito realizado encontra-se ilustrado na Figura 76 e alguns dos ensaios encontram-se listados na Tabela 10.

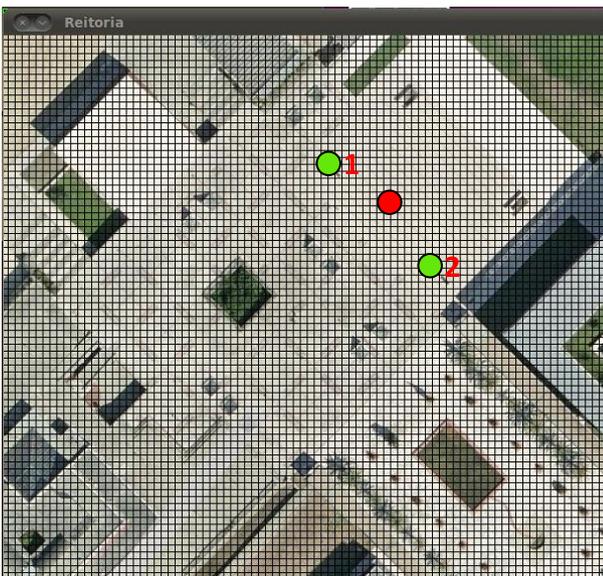


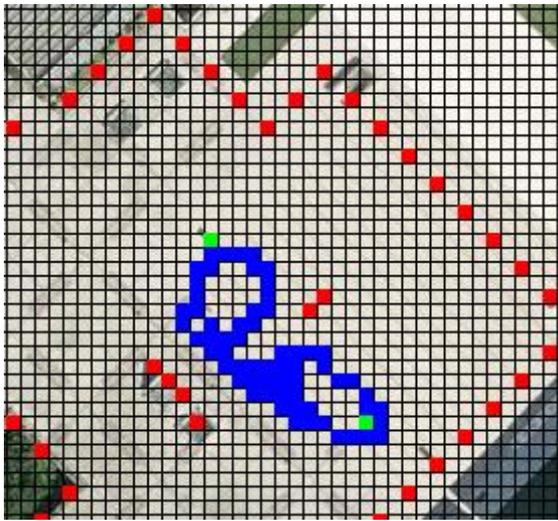
Figura 76 - Planeamento Matricial - Circuito do ensaio do tipo 3

Tabela 10 - Planeamento Matricial - Ensaio realizados do Tipo 3

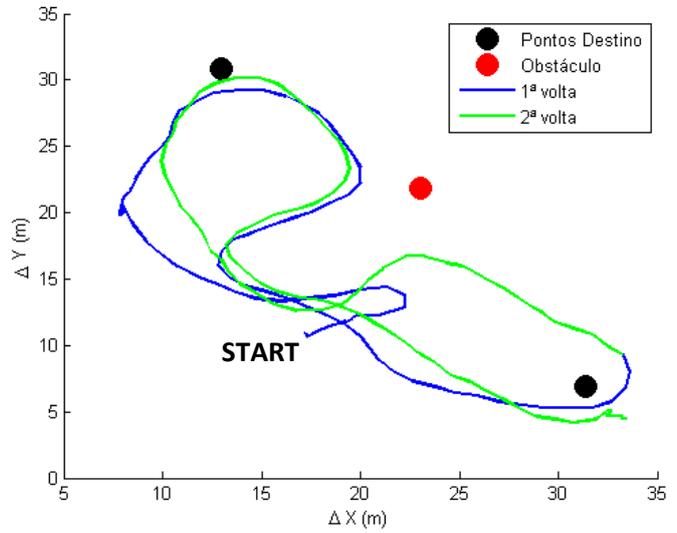
Ensaio	1	2	3	4
E (constante atrator)	1	1	10	1
N (constante repulsor)	10	20	20	10
Distância de influência (células)	4	3	5	4
Dimensão da célula (metros)	2x2	2x2	2x2	4x4



• **Resultados do ensaio do tipo 3:**



(a)

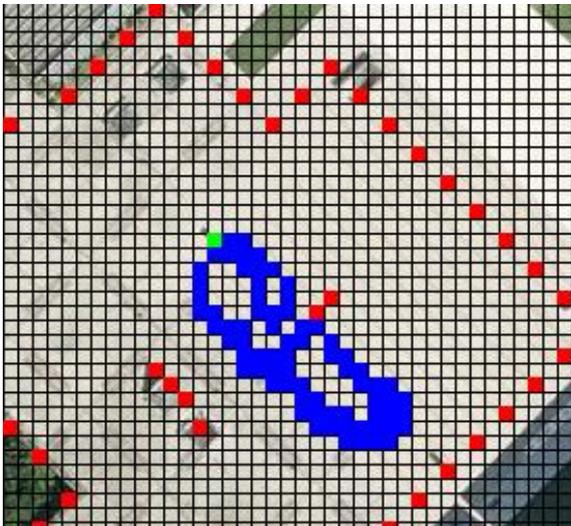


(b)

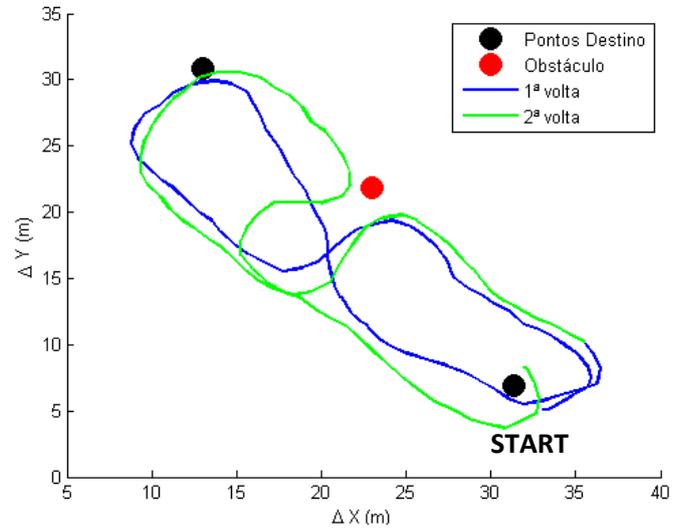
Figura 77 - Planejamento matricial - ensaio do tipo 3 – 1

(a) Representação no mapa (parte)

(b) - Representação das voltas realizadas



(a)

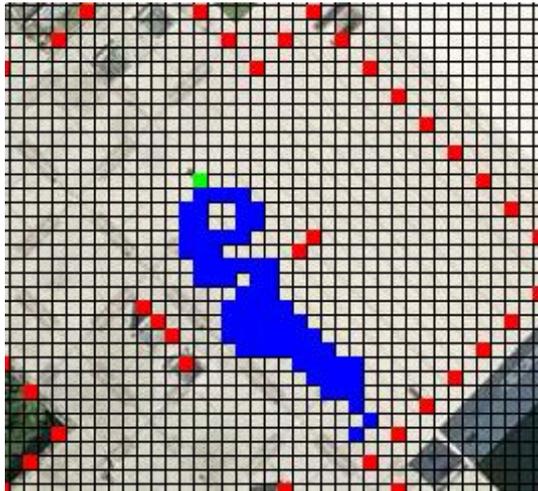


(b)

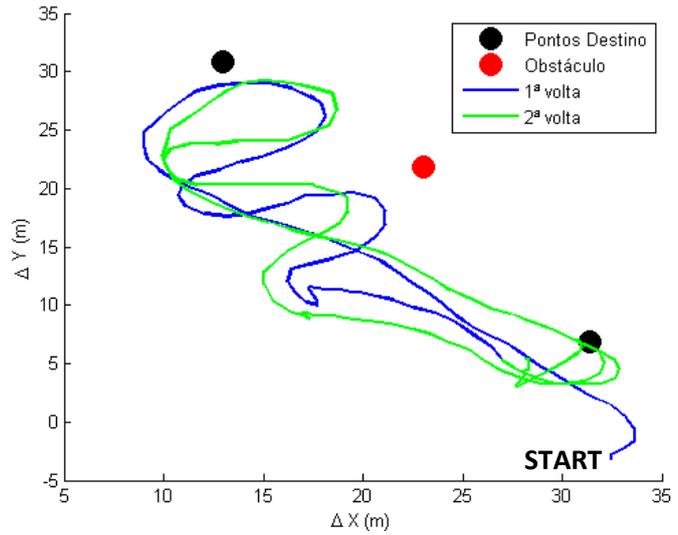
Figura 78 - Planejamento matricial - ensaio do tipo 3 – 2

(a) - Representação no mapa (parte)

(b) - Representação das voltas realizadas



(a)

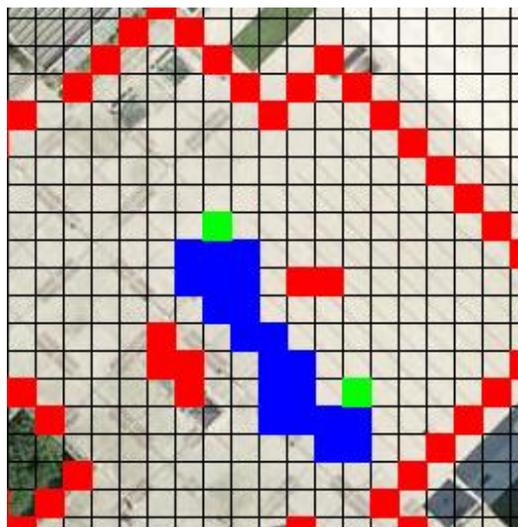


(b)

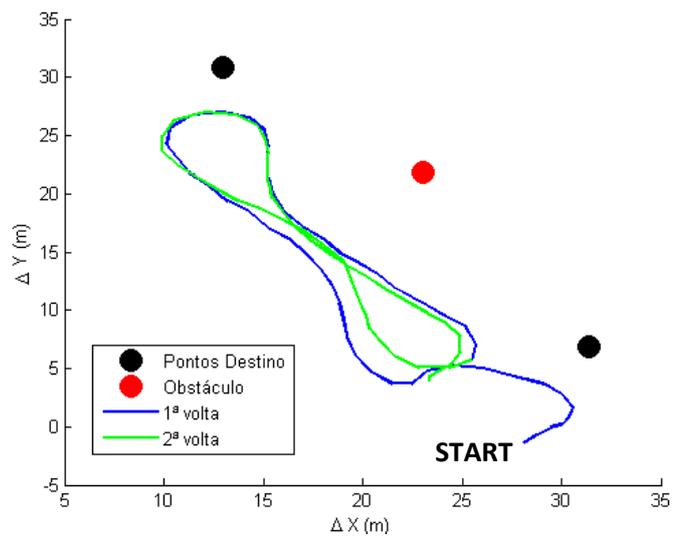
Figura 79 - Planeamento matricial - ensaio do tipo 3 – 3

(a) – Representação no mapa (parte)

(b) - Representação das voltas realizadas



(a)



(b)

Figura 80 - Planeamento matricial - ensaio do tipo 3 – 4

(a) – Representação no mapa (parte)

(b) - Representação das voltas realizadas

Com a realização destes ensaios verifica-se que os obstáculos virtuais influenciam, como esperado, a escolha de caminhos. Comparando os ensaios 3-1 (Figura 77) e 3-2 (Figura 78) é notável o impacto da alteração do valor da distância de influência, registando uma maior aproximação do robô ao obstáculo, antes de iniciar o seu desvio, no ensaio 3-2, onde a distância de influência foi reduzida para três células. No ensaio 3-3 (Figura 79) utilizou-se uma constante repulsora elevada e aumentou-se a distância de influência para cinco células, o conjunto destes parâmetros fez com que o robô percorresse um caminho muito serpenteado. Conclui-se que a junção destes dois parâmetros



faz com que a influência dos obstáculos seja muito tardia e muito forte, repulsando o robô de uma maneira muito intensa quando este se aproxima dele. Outro fator que provocou este caminho foi a repulsão dos obstáculos virtuais já existentes no mapa, criados para sinalizar alguns obstáculos reais, estáticos. Com uma distância de influência elevada, estes obstáculos são contabilizados no cálculo do campo de potencial. Realizou-se ainda um ensaio com células maiores, no ensaio 3-4 (Figura 80) usou-se células de 4x4 metros e registou-se um comportamento semelhante aos ensaios anteriores, em que os obstáculos influenciam o caminho realizado. A influência do obstáculo colocado em conjunto com os obstáculos virtuais existentes no mapa faz com que o caminho realizado tenha a forma de um “8”, tornando o robô vítima de forças de repulsão dos dois lados quando passa entre os obstáculos.

- **Ensaio do tipo 4**

Como já mencionado, os obstáculos detetados pelo laser são considerados obstáculos temporários e que, apenas influenciam o cálculo do campo potencial enquanto estão a ser detetados. Assim, decidiu-se realizar ensaios somente com obstáculos reais, detetados pelo laser, para analisar o comportamento do robô quando este deixa de detetar os obstáculos. Utilizou-se o circuito que se encontra na Figura 81, e usou-se de uma caixa de cartão como obstáculo que foi colocado na posição indicada. Os ensaios, mais conclusivos, realizados encontram-se listados na Tabela 11 e serão ilustrados nas páginas seguintes (73 e 74).



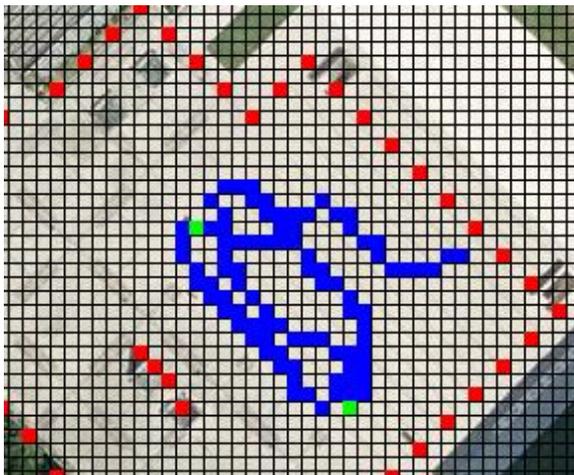
Figura 81 - Planeamento Matricial - Circuito do ensaio do tipo 4

Tabela 11 - Planeamento Matricial - Ensaio realizado do Tipo 4

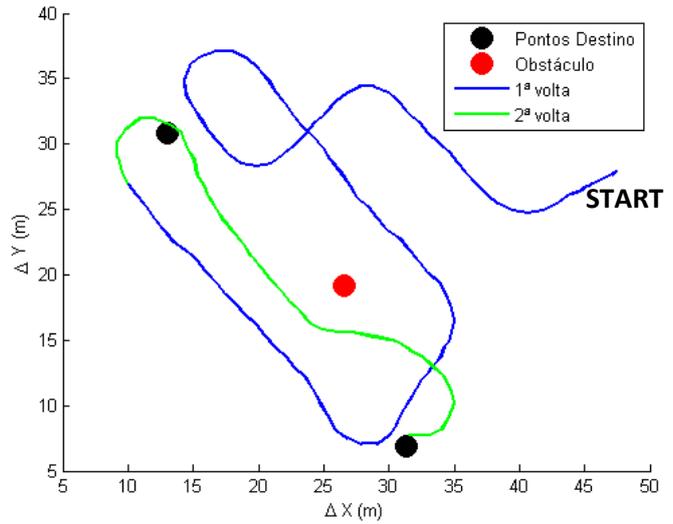
Ensaio	1	2	3
E (constante atrator)	1	1	10
N (constante repulsor)	10	10	20
Distância de influência (células)	4	3	5
Dimensão da célula (metros)	2x2	2x2	2x2



• **Resultados do ensaio de tipo 4:**



(a)

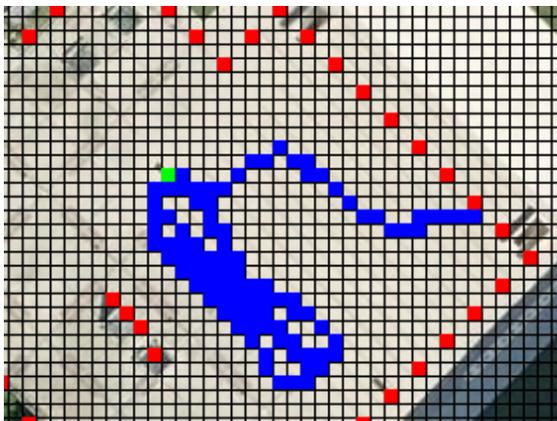


(b)

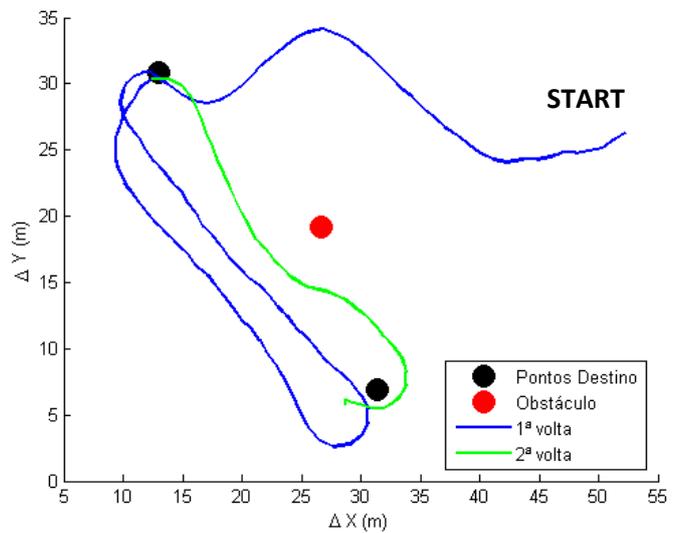
Figura 82 - Planeamento matricial - ensaio do tipo 4 –1

(a) – Representação no mapa (parte)

(b) - Representação das voltas realizadas



(a)

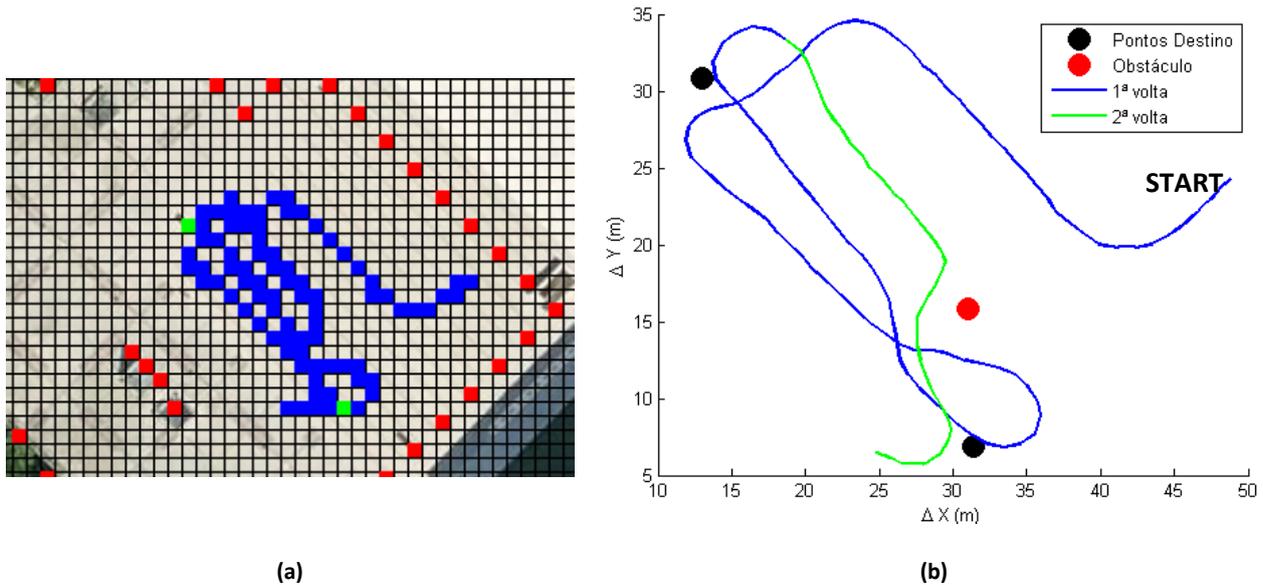


(b)

Figura 83 - Planeamento matricial - ensaio do tipo 4 – 2

(a) – Representação no mapa (parte)

(b) - Representação das voltas realizadas



(a) (b)
Figura 84 - Planejamento matricial - ensaio do tipo 4 – 3
(a) – Representação no mapa (parte) (b) - Representação das voltas realizadas

Com a realização destes ensaios verificou-se a influência dos obstáculos detetados pelo laser na escolha de trajetória. Nos três ensaios listados, 4-1 (Figura 82), 4-2 (Figura 83) e 4-3 (Figura 84), registaram-se resultados positivos, visto que a deteção pelo laser impulsionou a uma repulsão e desvio do robô. Um dos objetivos deste ensaio era avaliar a influência do obstáculo, quando deixa de ser detetado pelo laser, nestas situações verificou-se uma mudança súbita de trajetória. Este fenómeno é mais notável no ensaio 4-1 (Figura 82), na segunda volta, onde o robô vira repentinamente para a sua esquerda, quando deixa de detetar o obstáculo. Estes desvios repentinos são perfeitamente normais, visto que nestas situações o planeador não “sabe” que um obstáculo se encontra por perto e calcula a trajetória apenas pelos dados recebidos.

- **Ensaio do tipo 5**

Decidiu-se realizar um tipo de ensaio completo, colocando os pontos destino mais separados e com vários obstáculos entre eles, tanto obstáculos reais como virtuais. Optou-se colocar os dois pontos de destino quase nas extremidades do mapa, elevando circunstancialmente o grau de dificuldade de um ensaio deste género. As razões principais por este aumento são o número de obstáculos que se encontram na zona e a distância relativamente reduzida a que se encontram; na Figura 85 encontram-se as posições dos pontos destino utilizados. Na Tabela 12 encontram-se listados os parâmetros usados nos ensaios que serão descritos seguidamente.



Figura 85 - Planeamento Matricial - Circuito do ensaio do tipo 5

Tabela 12 - Planeamento Matricial - Ensaio realizados do Tipo 5

Ensaio	1	2
E (constante atrator)	1	1
N (constante repulsor)	10	10
Distância de influência (células)	5	3
Dimensão da célula (metros)	2x2	2x2

- Resultados do ensaio de tipo 5

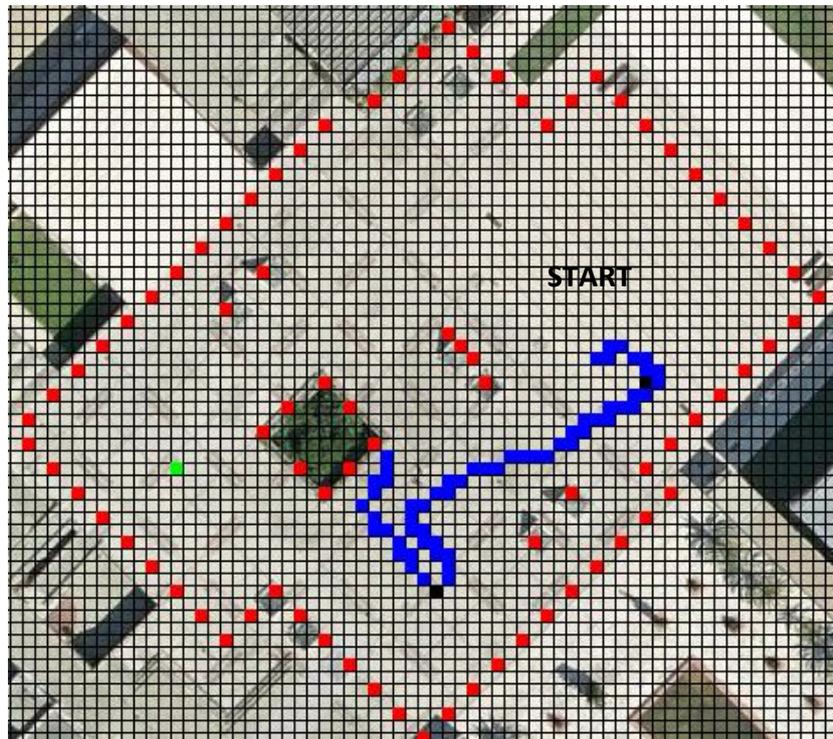


Figura 86 - Planeamento Matricial - Ensaio do tipo 5-1

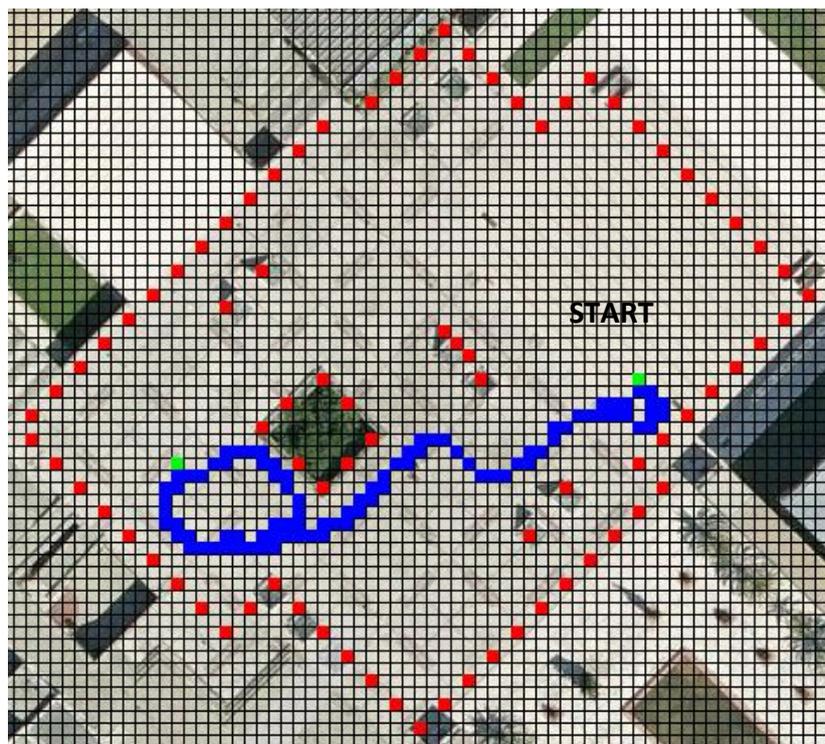


Figura 87 - Planeamento Matricial - Ensaio do tipo 5-2

Após muitos ensaios sem êxito conclui-se que este planeador não consegue efetuar caminhos extensos com muitos obstáculos entre os pontos objetivo, só se apresentou os resultados de dois ensaios, Figura 86 e Figura 87, visto que nenhum outro obteve melhores resultados. Nenhum dos ensaios realizados conseguiu chegar ao ponto destino, isto devido principalmente ao número elevado de obstáculos que se encontram na zona (banco, candeeiros, etc.). A existência de muitos obstáculos limita as opções de trajetória e muitas vezes leva o robô até um ponto mínimo local, deixando-o sem saída. Uma solução lógica para este problema seria aumentar o valor da distância de influência dos obstáculos, mas esta solução gera outros problemas que dificultam a realização do caminho. Ao elevar o valor da distância de influência o robô será influenciado por obstáculos que se encontram relativamente longes, e que impossibilitaram o robô de atingir o destino

5.3. Planeamento por Campos Potencial Radial

De modo a facilitar a comparação entre o planeamento matricial e o radial, decidiu-se realizar os mesmos tipos de ensaios e sobre as mesmas condições. Tal como nos ensaios anteriores, decidiu-se usar uma velocidade constante em todos os ensaios realizados de modo a diminuir o erro de posicionamento de GPS. De seguida encontram-se os ensaios realizados e os principais resultados obtidos.



- **Ensaio do tipo 1**

Para estes ensaios utilizou-se os mesmos pontos de destino do circuito ilustrado na Figura 68, tal como no planeador matricial este ensaio serve para avaliar o comportamento atrator dos pontos objetivo. Na Tabela 13 encontram-se listados os parâmetros usados na concretização destes ensaios.

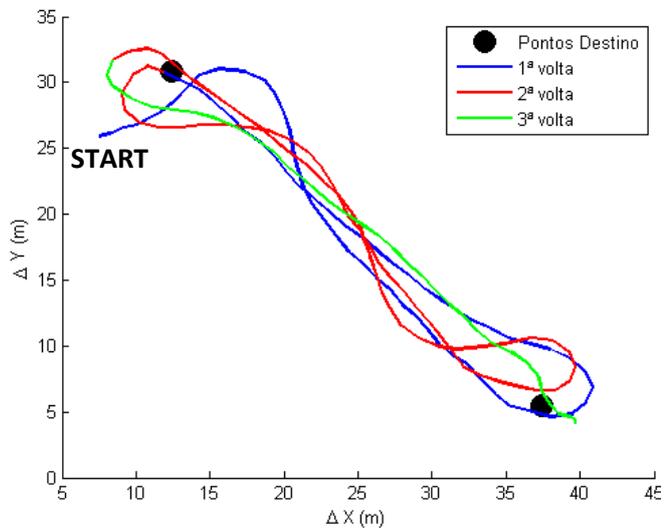
Tabela 13 - Planeamento Radial – Ensaio realizados do tipo 1

Ensaio	1	2
E (constante atrator)	10	1
N (constante repulsor)	20	20
Distância de Influência (metros)	7	7

- **Resultados do ensaio de tipo 1**



(a)



(b)

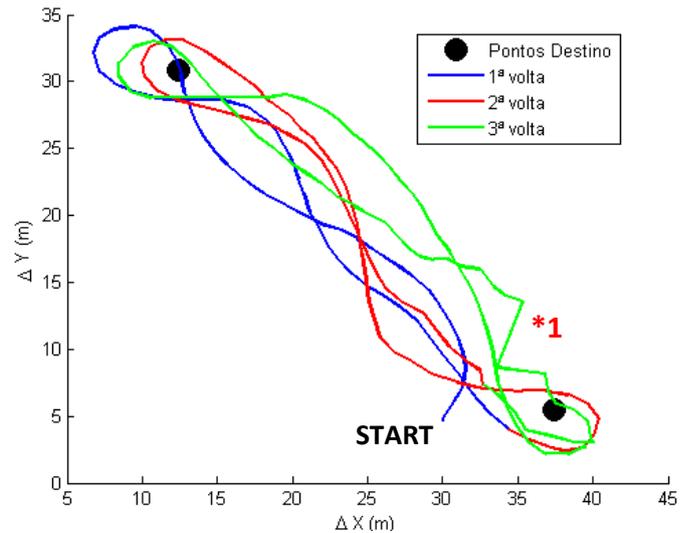
Figura 88 - Planeamento radial - ensaio do tipo 1 – 1

(a) – Representação no mapa (parte)

(b) - Representação das voltas realizadas



(a)



(b)

Figura 89 - Planeamento radial - ensaio do tipo 1 – 2

(a) – Representação no mapa (parte)

(b) - Representação das voltas realizadas

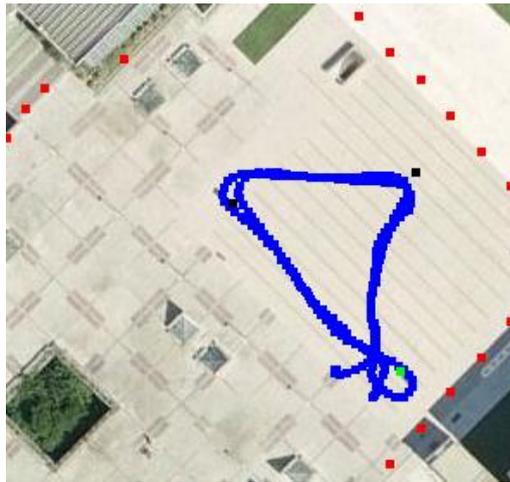
Com a realização destes ensaios obtiveram-se resultados semelhantes ao planeador matricial, em que o aumento da constante atratora torna os caminhos percorridos mais regulares. No ensaio 1-1 (Figura 88) observaram-se caminhos mais compactos e regulares, enquanto que no ensaio 1-2 (Figura 89) verifica-se a realização de caminhos mais dispersos. É de salientar que na zona marcada com “*1” na Figura 89 ocorreu um desvio no caminho que não aconteceu na realidade, este apenas está representado devido a um erro de leitura efetuado pelo GPS.

• **Ensaio do tipo 2**

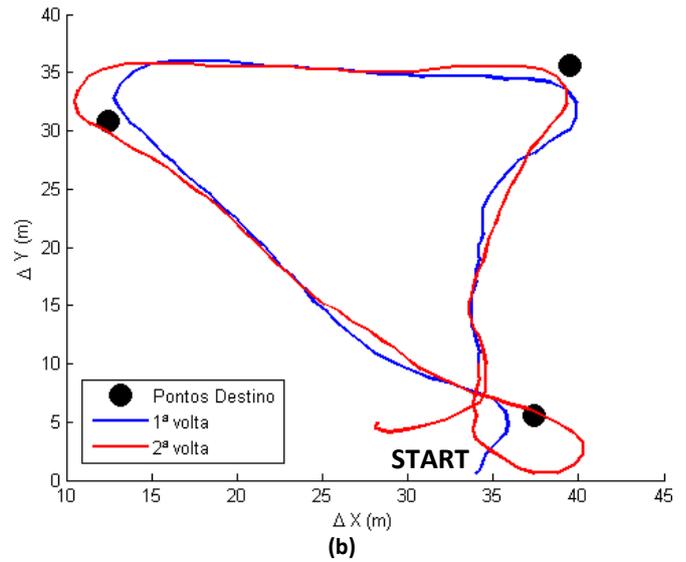
Neste ensaio usou-se o mesmo circuito do ensaio do tipo 2 do planeamento matricial, ilustrado na Figura 72, constituído por um circuito simples com três pontos destino; tal como no planeador matricial, este ensaio tem como objetivo avaliar o comportamento do robô em meios sem obstáculos. Assim decidiu-se alterar apenas a constante atratora nos ensaios realizados, os referidos encontram-se listados na Tabela 14 e os seus resultados encontram-se ilustrados de seguida.

Tabela 14 - Planeamento Radial - Ensaio realizados do tipo 2

Ensaio	1	2
E (constante atrator)	1	20
N (constante repulsor)	10	10
Distância de influência (metros)	7	7



(a)



(b)

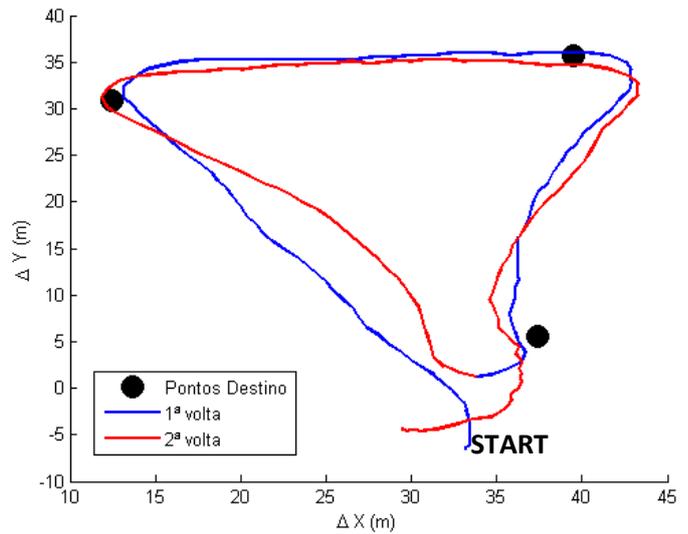
Figura 90 - Planeamento radial - ensaio do tipo 2 – 1

(a) – Representação no mapa (parte)

(b) - Representação das voltas realizadas



(a)



(b)

Figura 91 - Planeamento radial - ensaio do tipo 2 – 2

(a) – Representação no mapa (parte)

(b) - Representação das voltas realizadas

Os resultados obtidos neste circuito, com este planeamento, são melhores que os ensaios realizados pelo planeamento matricial no mesmo circuito. Comparando os ensaios 2-1 (Figura 90) e 2-2 (Figura 91) do planeamento radial com os resultados obtidos com planeamento matricial (Figura 73 e Figura 74), é bem visível uma melhoria dos caminhos percorridos. Nestes ensaios os caminhos percorridos são mais diretos ao destino e não realizam trajetórias serpenteadas, a principal razão por esta melhoria de resultados é o aumento de direções possíveis que este planeador oferece. Este planeador calcula o valor de potencial para 16 pontos à sua volta, em que 11 encontram-se apenas na parte dianteira do robô, oferecendo assim, uma maior oferta de destino locais ao robô.



- **Ensaio do tipo 3**

Tal como no ensaio tipo 3 do planeamento matricial, este tipo de ensaio também tem como objetivo avaliar o comportamento do robô com a interação de obstáculos virtuais no seu caminho. Utilizou-se o mesmo percurso que no planeamento anterior, ilustrado na Figura 76, colocou-se um obstáculo virtual entre os dois pontos de destino e realizaram-se ensaios com parâmetros diferentes de modo a verificar a influência de cada um destes na escolha de trajetórias. Alguns dos ensaios realizados encontram-se listados na Tabela 15 e os seus resultados encontram-se ilustrados de seguida.

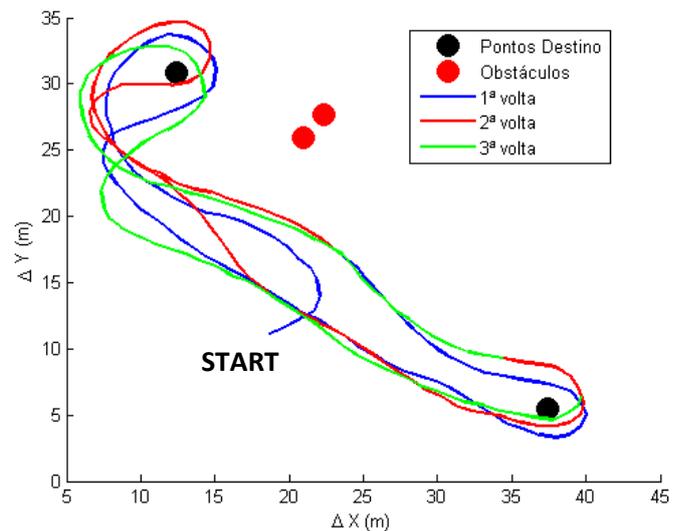
Tabela 15 - Planeamento Radial - Ensaios realizados do tipo3

Ensaio	1	2	3
E (constante atrator)	1	10	1
N (constante repulsor)	10	20	20
Distância de influência (metros)	5	7	3

- **Resultados do ensaio do tipo 3**



(a)



(b)

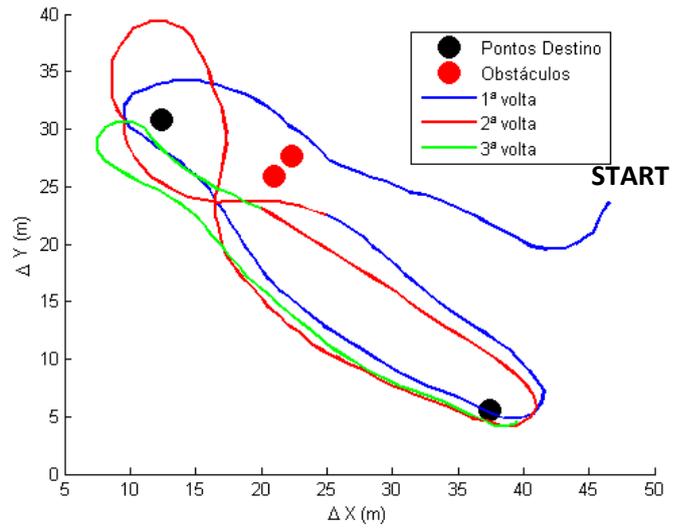
Figura 92 - Planeamento radial - ensaio do tipo 3 –.1

(a) – Representação no mapa (parte)

(b) - Representação das voltas realizadas



(a)



(b)

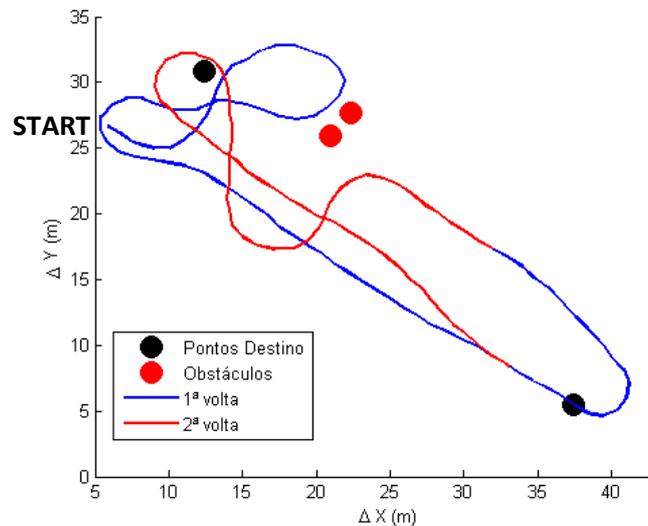
Figura 93 - Planeamento radial - ensaio do tipo 3 – 2

(a) – Representação no mapa (parte)

(b) - Representação das voltas realizadas



(a)



(b)

Figura 94 - Planeamento radial - ensaio do tipo 3 – 3

(a) – Representação no mapa (parte)

(b) - Representação das voltas realizadas

Este tipo de ensaio tinha como objetivo analisar o comportamento do robô com a interação de obstáculos virtuais, e após a realização conclui-se que os obstáculos influenciam, da maneira esperada, a escolha de caminho. Nos três ensaios demonstrados, 3-1 (Figura 92), 3-2 (Figura 93) e 3-3 (Figura 94), são perceptíveis as consequências das alterações dos valores dos parâmetros de planeamento. No ensaio 3-2 (Figura 93) ocorre uma aproximação do robô ao obstáculo; isto é causado pela grandeza dos valores das constantes atractora e repulsora, que são ambos elevados e



bastante próximos, causando um nível de atração quase semelhante ao de repulsão que, em consequência, aproxima muito o robô ao obstáculo por este se encontrar no caminho do destino. Conclui-se assim que é preferível estabelecer valores de grandeza superior para a constante de repulsão de modo a não ocorrerem grandes aproximações do robô aos obstáculos. No ensaio 3-3 (Figura 94), deparamos com uma nova aproximação do robô ao obstáculo, mas esta aproximação é consequência do valor da distância de influência ser reduzida.

- **Ensaio do tipo 4**

Tal como no planeamento matricial decidiu-se realizar ensaios com um obstáculo real entre os dois pontos destino de modo poder avaliar o comportamento do robô quando deteta e deixa de detetar um obstáculo. Colocou-se uma caixa de cartão na posição ilustrada Figura 95, e realizaram-se vários ensaios deste tipo, os mais conclusivos encontram-se listados na Tabela 16 e ilustrados de seguida.



Figura 95 - Planeamento Radial - Circuito do ensaio do tipo

3

Tabela 16 - Planeamento Radial - Ensaio realizados do Tipo 4

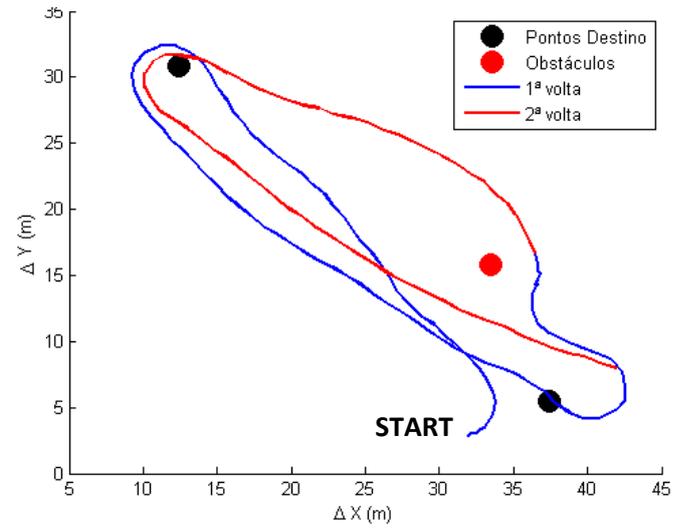
Ensaio	1	2	3
E (constante atrator)	1	1	10
N (constante repulsor)	20	10	20
Distância de influência (metros)	3	5	6



- **Resultados do ensaio do tipo 4**



(a)



(b)

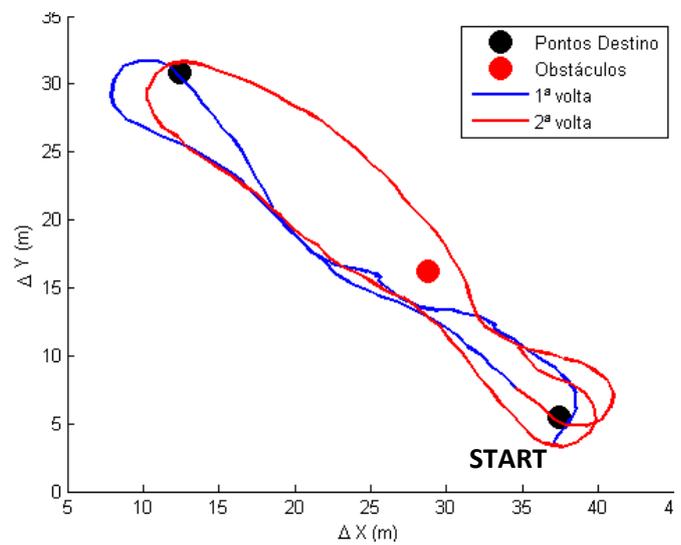
Figura 96 - Planeamento radial - ensaio do tipo 4 – 1

(a) – Representação no mapa (parte)

(b) - Representação das voltas realizadas



(a)



(b)

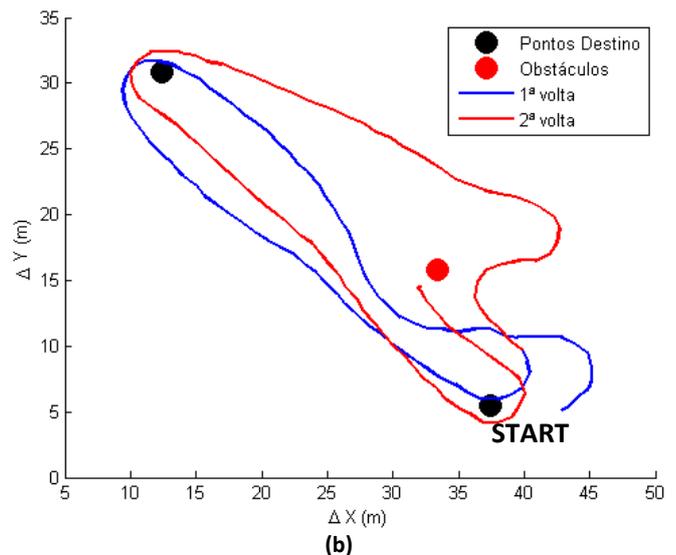
Figura 97 - Planeamento radial - ensaio do tipo 4 – 2

(a) – Representação no mapa (parte)

(b) - Representação das voltas realizadas



(a)



(b)

Figura 98 - Planeamento radial - ensaio do tipo 4 – 3

(a) - Representação no mapa (parte)

(b)- Representação das voltas realizadas

Nos três ensaios registados, ensaios 4-1 (Figura 96), 4-2 (Figura 97) e 4-3 (Figura 98), obteve-se bons resultados e é notável a influência de cada parâmetro na concretização dos caminhos. No ensaio 4-2 (Figura 97) é facilmente observável, em comparação com os outros dois ensaios, que a diminuição da constante repulsora reduz a “força” com que o robô é repellido pelo obstáculo, obtendo-se assim caminhos que passam mais próximos do obstáculo. No ensaio 4-3 (Figura 98) ocorre um instante em que o robô se encontra diretamente direcionado para obstáculo e sofre um forte desvio para a sua direita, causando um contorno exagerado ao obstáculo. Este fenómeno ocorre devido ao número de obstáculos que são contabilizados no cálculo do campo repulsor, na realidade o obstáculo é apenas um, mas o laser deteta obstáculos pontuais e não corpos inteiros. Para diminuir este efeito inseriu-se um filtro na captação de obstáculos pelo laser, contabilizando apenas pontos que encontrem a uma certa distância entre eles.

- **Ensaio do tipo 5**

Para finalizar a análise do comportamento deste planeador decidiram-se realizar ensaios com pontos de destino mais afastados e com mais obstáculos entre si, pré-definidos e detetados pelo laser. Utilizaram-se os mesmos pontos de destino do ensaio final do planeador matricial, ilustrados na Figura 85. De seguida serão apresentados alguns dos resultados obtidos, na Tabela 17 encontram-se listados os parâmetros utilizados dos ensaios apresentados.



Tabela 17 – Planeamento Radial – Ensaios realizados do tipo 5

Ensaio	1	2	3	4
E (constante atrator)	1	5	5	1
N (constante repulsor)	10	10	10	10
Distância de influência (metros)	8	8	10	5

Resultados do ensaio do tipo 5

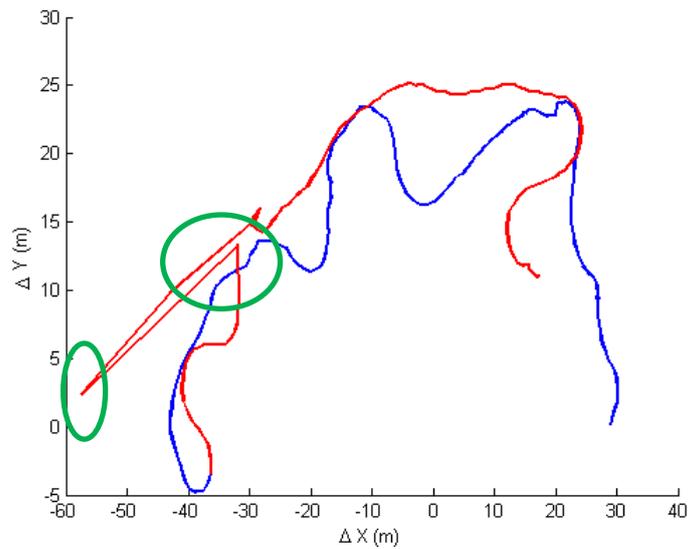


Figura 99 - Planeamento radial - ensaio do tipo 5 - 1.a

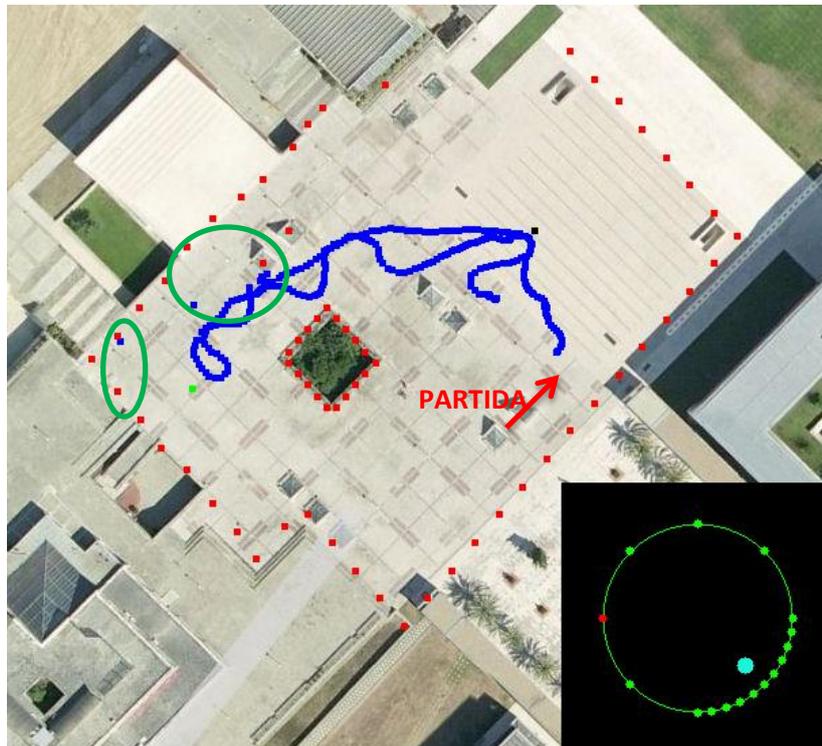


Figura 100 - Planeamento radial - ensaio do tipo 5 – 1.b

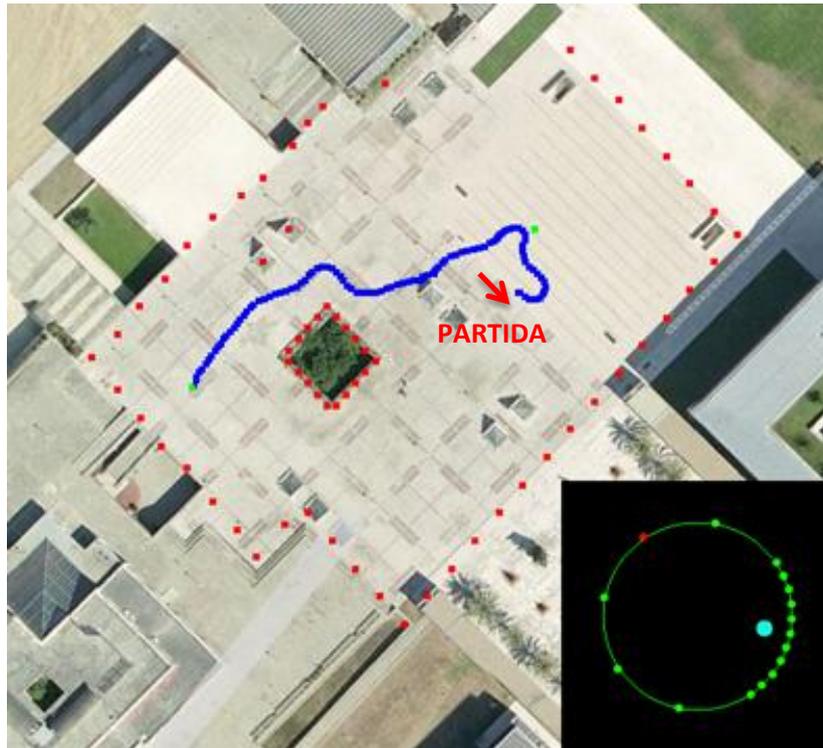


Figura 101 - Planeamento radial - ensaio do tipo 5 – 2

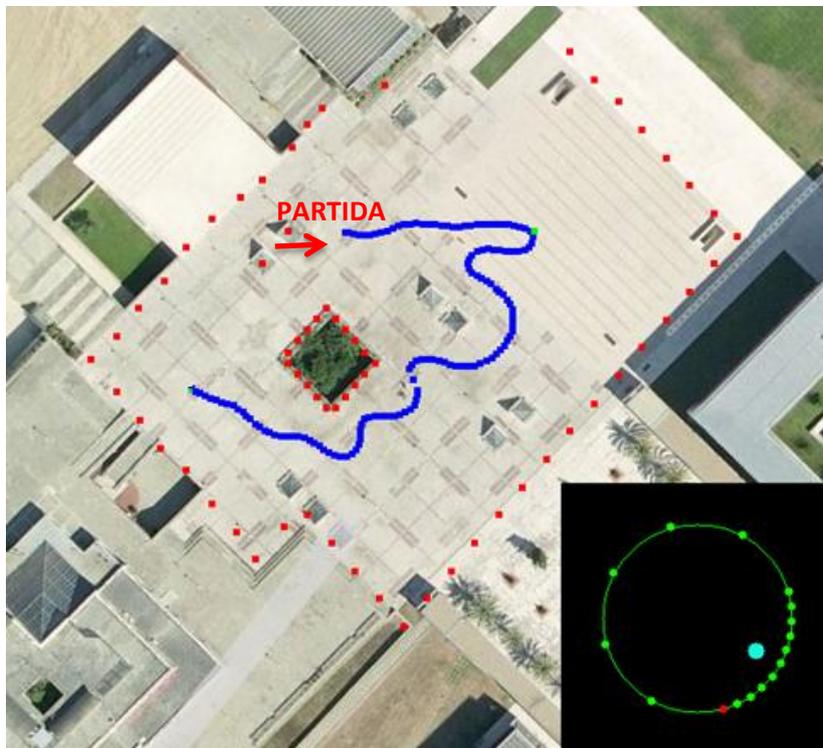


Figura 102 - Planeamento radial - ensaio do tipo 5 – 3

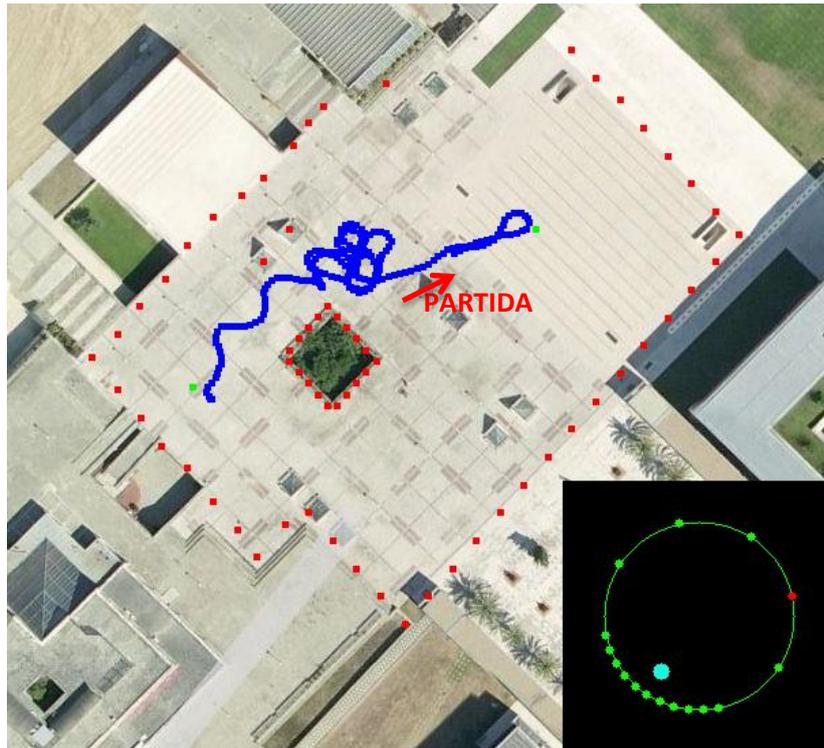


Figura 103 - Planeamento radial - ensaio do tipo 5 – 4

O ensaio do tipo 5 do planeamento radial apresenta resultados positivos que demonstram a eficácia do método implementado, comprovam que com este planeador o robô consegue movimentar-se entre duas localizações distantes de maneira segura e contínua, desviando-se de potenciais obstáculos que possam aparecer no seu caminho. No primeiro ensaio realizado, representado na Figura 99 e Figura 100, registou-se mais uma vez um grande erro de leitura por parte do recetor GPS, mas como este é apenas momentâneo não chegou a influenciar a escolha de trajetória do robô. Os erros de leitura de posição referidos estão marcados por círculos verdes nas figuras mencionadas. Neste ensaio deixou-se o robô dar uma volta completa ao circuito constituído pelos dois pontos destino pré-definidos, este foi percorrido com sucesso e sem ocorrência de algum incidente. Nos ensaios restantes, representados na Figura 101, Figura 102 e Figura 103, decidiu-se realizar apenas o caminho necessário para ligar os dois pontos destino definidos. Em cada ensaio efetuou-se a alteração dos parâmetros de navegação de modo a avaliar a influência de cada um na escolha de caminhos, e para avaliar a robustez do planeador os ensaios foram iniciados sempre de localizações diferentes. Todos os ensaios apresentados conseguiram atingir os destinos com sucesso e com nenhuma ocorrência de embate com algum obstáculo. No último ensaio realizado, Figura 103, o robô andou às voltas entre o primeiro e segundo ponto destino, fenómeno devido a dois fatores. Primeiro, ao alcance do sensor laser ser de apenas 5,6 metros, e segundo, ao facto de haver muitos obstáculos, candeeiros e outras estruturas, na zona em questão. Assim, o planeador apenas



contabiliza os obstáculos que estão no seu alcance, podendo ser encaminhado para uma zona com outros obstáculos originando num novo desvio de trajetória. A constante detecção de obstáculos resulta na realização dos caminhos circulares até ser encontrado um caminho seguro para se aproximar ao destino.

O algoritmo desenvolvido não é 100% eficaz, por vezes encaminha o robô para posições de mínimos locais e até mesmo uma escolha errada de trajetória. A escolha errada apenas ocorre quando sucede um erro de leitura por parte do GPS, erros de localização e orientação. Os últimos ocorrem principalmente em situações em que o robô se encontra a realizar caminhos circulares, curvas, com raios reduzidos. Situações que provocam grandes alterações na orientação verdadeira do robô mas pequenas na localização do mesmo. Este fator pode levar à obtenção de uma orientação não correta por parte do recetor GPS, visto que a orientação provinda na mensagem do GPS é calculada através das coordenadas geográficas de dois pontos passados. Se a curva realizada for de raio muito reduzido, pode ocorrer uma variação insuficiente de coordenadas GPS para ser calculado uma orientação correta do robô. O erro de posição do próprio GPS também influencia a obtenção da orientação nas situações descritas. Durante a realização destes ensaios observou-se a ocorrência dos erros mencionados.



6. Conclusões e Trabalhos Futuros

Os principais objetivos do trabalho eram efetuar as alterações necessárias, tanto a nível de *hardware* como a nível de *software*, para implementar a capacidade de navegação, por condução autónoma, do robô de série Atlas 2000 em meios exteriores. A inserção de um sistema de monitorização das baterias do robô era um objetivo secundário a ser realizar ao longo deste trabalho. Neste capítulo será feita uma análise conclusiva sobre o trabalho desenvolvido, mencionando os objetivos cumpridos e os que ficaram aquém das expectativas. Por fim serão mencionadas propostas para trabalhos futuros com novas funcionalidade que seriam interessantes implementar no sistema atual.

6.1. Conclusões

As intervenções mecânicas para possibilitar o deslocamento em meios exteriores foram uma tarefa mais desafiante do que estava inicialmente prevista, especialmente pela dificuldade em obter rodas que satisfizessem todos os requisitos desejados. Após a sua obtenção e instalação, verificou-se uma melhoria notável no deslocamento do robô em terrenos mais irregulares, conseguindo-se obter a elevação do chassis em relação ao solo desejada, sendo possível afirmar que o objetivo foi conseguido.

A migração do *software* para a filosofia CARMEN/IPC foi uma tarefa delicada que contou com a participação de outros elementos da equipa Atlas, resultando numa migração com sucesso. Assim, o programa responsável pela condução autónoma do robô foi dividido em vários módulos que comunicam entre si, realizando trocas de mensagens contendo o resultado do processamento do respetivo módulo. A filosofia modular oferece várias vantagens, sendo uma das mais importantes a fácil edição de código responsável por determinada tarefa sem influenciar código de outras tarefas. Graças ao sistema modular, a inserção de novo código, responsável por novas tarefas, torna-se numa operação mais organizada e simples, sendo apenas necessário criar os módulos desejadas e



estabelecer as comunicações entre eles. Como se pôde verificar com a implementação dos vários módulos criados ao longo deste trabalho para a condução autónoma por GPS.

A implementação de um sistema de planeamento de caminhos por GPS foi a tarefa mais desafiante deste trabalho, especialmente o desenvolvimento do *software* necessário para a condução autónoma. A instalação do *hardware*, o recetor GPS, foi uma tarefa relativamente simples sendo apenas concebido um suporte de fixação em acrílico e uma placa de circuito impresso responsável pela sua alimentação, a partir da placa mãe do robô, e pela conversão e envio de dados recebidos pelo recetor. Após a sua instalação, desenvolveu-se um módulo dedicado à obtenção e separação dos dados GPS, tarefa facilitada pelo recetor utilizar o protocolo NMEA 0183. Realizaram-se dois módulos de planeamento de caminhos por campos de potencial, o matricial, que analisa e divide todo o espaço em estudo em sectores independentes, calculando o campo de potencial para cada um destes. E o radial, que apenas calcula o campo de potencial para uma série de células que se encontram numa zona próxima em torno do robô. Um dos maiores problemas encontrados durante o desenvolvimento dos módulos de condução autónoma por campos de potencial foi a obtenção de uma orientação (*True Course*) válida. Visto que esta provém do recetor GPS, por vezes, com algum erro. A orientação do robô é um dos fatores mais importantes do planeamento de caminhos, se esta estiver incorreta pode levar a uma escolha incorreta de caminho.

O planeador matricial proposto obtém bons resultados em meios com poucos obstáculos entre os pontos de destino, isto deve-se à sua natureza de comparação por matriz e ao facto de um obstáculo “ocupar” uma célula inteira da matriz. Assim, um elevado número de obstáculos numa zona aproximada pode originar uma situação em que o algoritmo não seja capaz de descobrir um caminho seguro e contínuo até ao destino global desejado. Podendo-se afirmar que o planeador por campos de potencial matricial foi bem desenvolvido e que, apesar das suas limitações, consegue obter resultados positivos em meios com poucos obstáculos. Registou-se melhores resultados com células de dois metros, com uma distância de influência de três a quatro células (seis a oito metros) utilizando uma constante repulsora dez vezes maior que a constante atratora. Pode-se concluir que os objetivos para este planeador foram alcançados.

O novo planeador desenvolvido, o radial, obtém bons resultados em quase todas as situações testadas, como se pode verificar nos resultados demonstrados no capítulo anterior. Como referido, o planeador não contém a capacidade de resolver todas as situações possíveis, podendo encontrar zonas de mínimos locais ou até ocorrer um erro de escolha de trajetória devido a erros de leitura de dados GPS. Mas globalmente registou-se bons resultados, especialmente nos ensaios utilizando constantes repulsoras superiores às atractoras e uma distância de influência de oito metros. Pode-se afirmar que com este planeador obtém-se resultados mais satisfatórios que o matricial e que os



objetivos iniciais deste trabalho foram alcançados. Ainda podendo-se afirmar que o algoritmo desenvolvido possui grande potencial para ser implementado em sistemas de tamanho real num futuro próximo. Em conclusão, o planeador desenvolvido satisfaz os objetivos inicialmente traçados, visto que, tendo as suas coordenadas GPS, consegue realizar caminhos entre duas localizações de maneira segura e contínua. O algoritmo desenvolvido implementa uma nova abordagem ao método de planeamento por campos de potencial, onde apenas são considerados pontos de interesse na zona mais próxima em torno do robô. Com uma diminuição do número de pontos de interesse a serem calculados, o tempo de computação do campo de potencial é diminuído, tornando este planeador mais leve que o matricial. Além de menos tempo de computação, este planeador ainda possui uma maior oferta de destinos locais garantindo melhores resultados na escolha de caminhos.

A incorporação de um sistema de monitorização das baterias foi um objetivo secundário deste trabalho, e o sistema concebido possui algum erro de leitura do valor de carga das baterias. Mas as leituras de que é capaz são suficientes para incluí-lo, num futuro próximo, no planeamento de caminhos, alertando o sistema, e o utilizador, quando alguma das baterias esteja a atingir um nível baixo.

Globalmente pode-se afirmar que os objetivos deste trabalho foram alcançados e que o trabalho desenvolvido tem potencial para ser continuado em trabalhos futuros.

6.2. Trabalhos Futuros

Como vista a resolver alguns problemas encontrados ou à adição de novas funcionalidades apresenta-se as seguintes tarefas:

- Melhoria do sistema de transmissão do motor que apresenta forte desgaste em alguns dos seus componentes;
- Substituição dos semieixos de transmissão;
- Inserção de uma bússola de modo a obter em tempo real a orientação real do robô, este sensor irá melhor consideravelmente o desempenho dos planeadores criados ao longo deste projeto;
- Colocação de sensores na retaguarda do robô para permitir manobras que incluam a marcha atrás;
- Implementação de sistemas ou funções para evitar os mínimos locais;
- Melhoramento e inclusão do sistema de monitorização das baterias no planeamento de caminhos;
- Inserção de mais um sensor de GPS de modo a diminuir o erro de localização;



- Implementação de mais mapas/zonas de planeamento de caminhos;



7. Referências

1. Silva, J. & J. Gomes. *ATLAS - Robot Móvel*. 2003 Outubro de 2011]; disponível em: <http://www2.mec.ua.pt/activities/graduationprojects/graduationprojectpages/2002-2003/robotatlas-josegomesjosesilva/>
2. Andrade, B. (2010). *Reconfiguração do robô ATLAS para implementação de unidade anti-colisão*. Aveiro: Universidade de Aveiro.
3. *Robotica 2011 – Festival Nacional de Robotica- Regulamento Geral* 2011 Outubro de 2011]; disponível em: http://robotica2011.ist.utl.pt/docs/2011_Regulamento_Geral-pt.pdf
4. Neta, M.O.M. *ATLAS II*. 2004 Outubro de 2011]; disponível em: <http://www2.mec.ua.pt/activities/graduationprojects/graduationprojectpages/2003-2004/RobotAtlasII-MiguelOliveiraMiguelNeta/paginas/index.htm>
5. Hokuyo Automatic Co., Ltd, *URG-04LX-UG01 – Specifications* Outubro de 2011]; disponível em http://www.hokuyo-aut.jp/02sensor/07scanner/download/data/URG-04LX_UG01_spec.pdf
6. CARMEN, Carnegie Mellon Navigation Toolkit, <http://carmen.sourceforge.net/> , Novembro de 2011.
7. R.Simmons and D. Apfelbaum, *A task description language for robot control*, 1998 Conference on Intelligent Robots and Systems (IROS), Victoria, CA.
8. M. Oliveira, P. Stein, J. Almeida, V. Santos, *Modular Scalable Architecture for the Navigation of the ATLAS Autonomous Robots*, Departamento de Engenharia Mecânica, TEMA – Universidade de Aveiro, 2009.
9. Kaplan, Elliott D.. *Understanding GPS – Principles and Applications*, Artech House Publishers, 2005.
10. De Lima Ottoni, Guilherme. *Planeamento de Trajectórias para Robôs Móveis*, Engenharia de Computação, Fundação Universidade Federal do Rio Grande, 2000.
11. Latombe, Jean-Claude. *Robot Motion Planning*, Kluwer Academic Publishers, 1991.
12. Santos, Vitor. *Apontamentos de Robótica Autónoma e Móvel*, Universidade de Aveiro, 2004



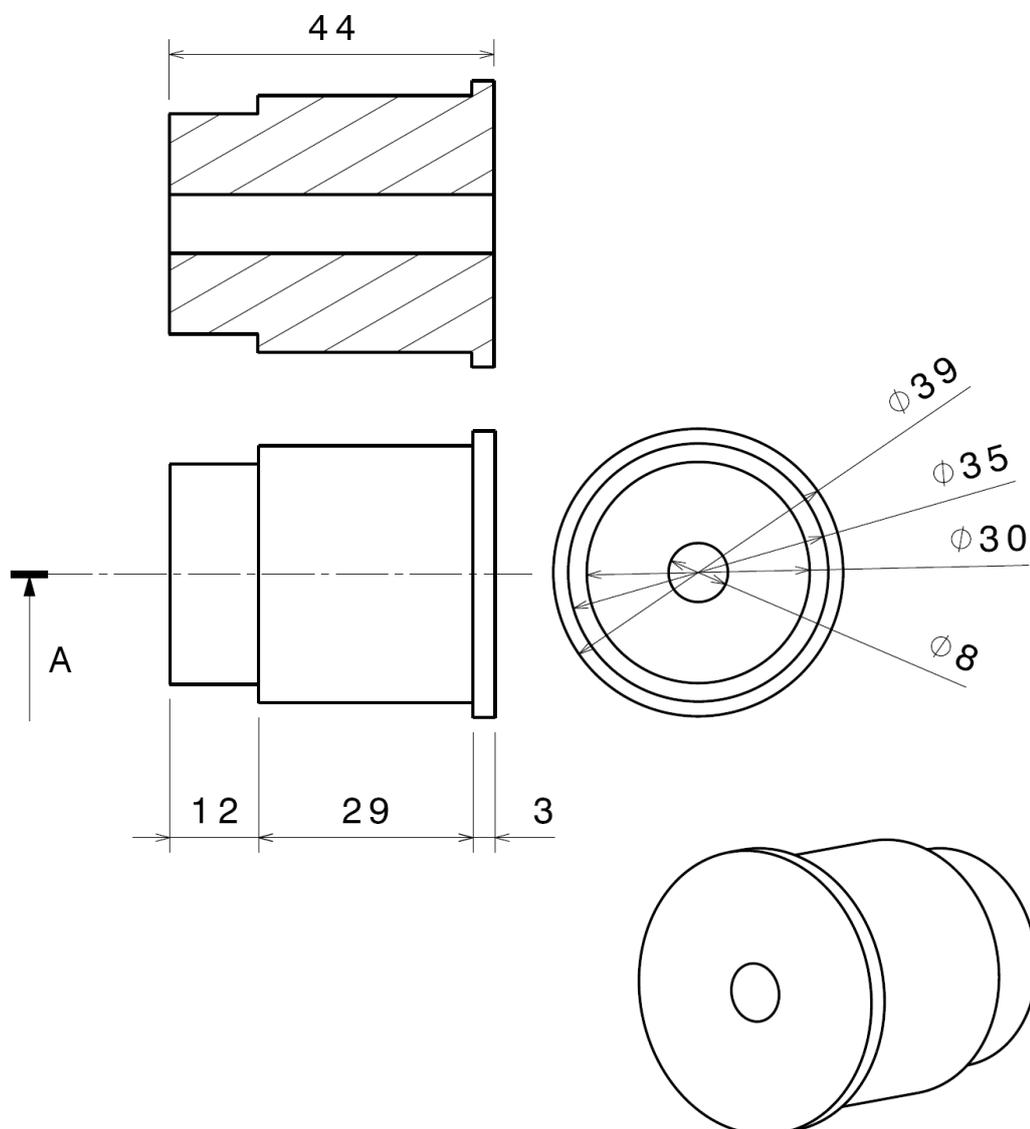
13. Goodrich, Micheal A.. *Potencial Fields Tutorial*, Outubro de 2011]; disponível em: http://borg.cc.gatech.edu/ipr/files/goodrich_potential_fields.pdf
14. Feo, Thomas & Resende Mauricio, *Greedy Randomized Adaptive Search Procedures*, Outubro 1994.
15. *Product User Manuel GPS Receiver Engine Board EM-406A*. GlobalSat Technology Corporation, Novembro 2011.
16. Libao, Open Source Audio Output Library, <http://www.xiph.org/ao/>, Dezembro de 2011.
17. Oliveira M., Santos V.. *Percepção Activa de Multi-camãra com perspectiva de imagem variável para navegação de robôs móveis*, 8ª Conferência de robôs moveis e competições, Festival Nacional de Robótica, Aveiro, Abril 2008.
18. Vítor Cancela Meireles, *Circuitos Eléctricos*, Lidel -4ª edição
19. R.W. Sinnott, *Virtues of the Haversine*, Sky and Telescope, vol. 68, no. 2, 1984, p. 159.
20. Hashim. Sani & Lu. Tien-Fu, *A New Strategy in Dynamic Time-dependent Motion Planning for Noholonomic Mobile Robots*, International Conference On Robotics and Biomimetics, China, 2009



8. Anexos

8.1. Anexo A

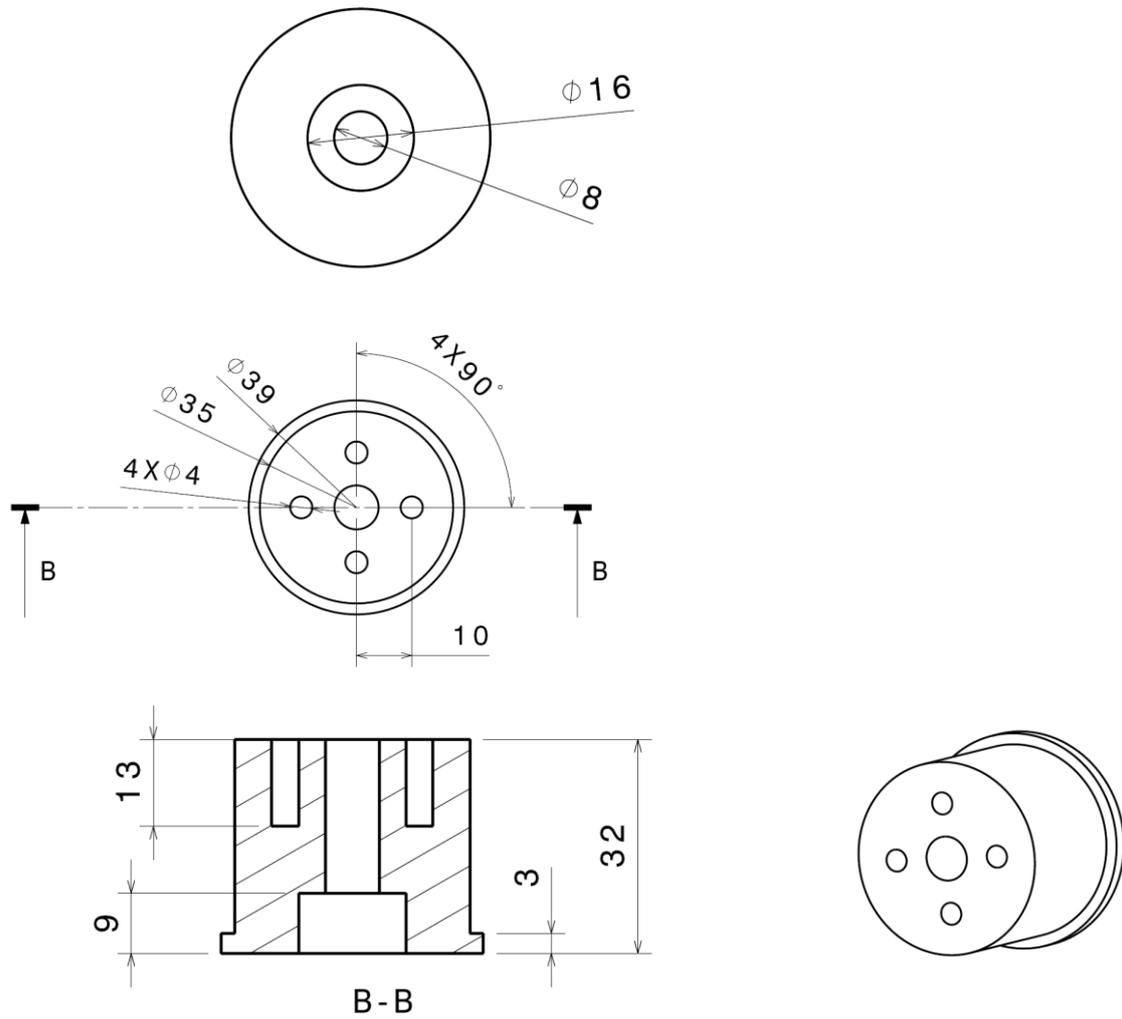
Desenhos de definição dos adaptadores das rodas dianteiras





8.2. Anexo B

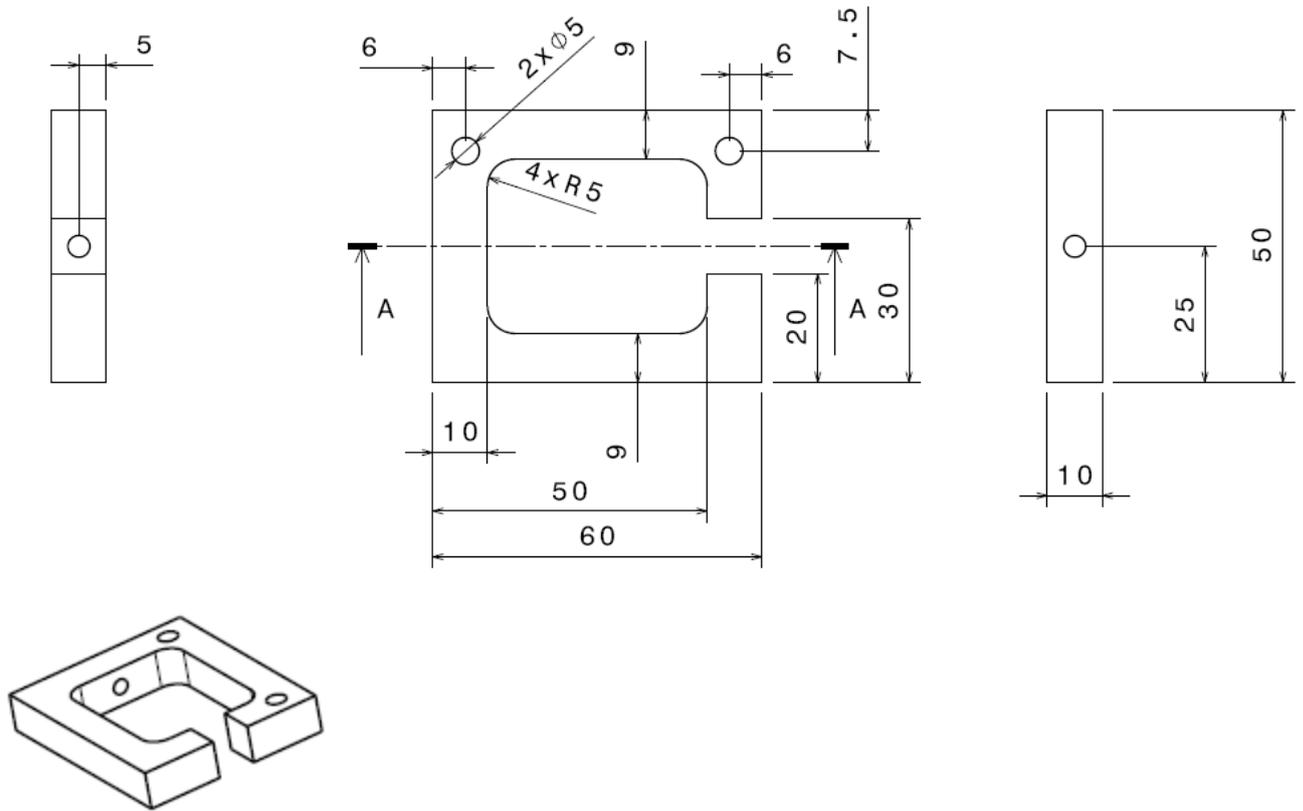
Desenhos de definição dos adaptadores das rodas traseiras





8.3. Anexo C

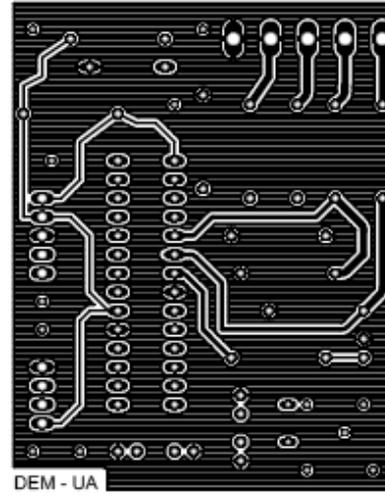
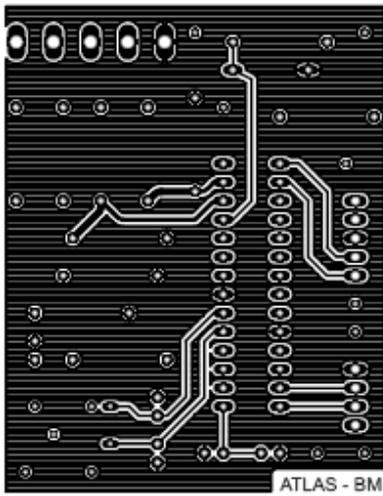
Desenhos de definição do suporte do recetor GPS





8.4. Anexo D

Placa do sistema de monitorização das baterias



8.5. Anexo E

Placa do sistema do recetor GPS

