



Pedro Miguel Ribeiro
Bouça Nova

Localização e Navegação Global do ATLASCAR2
usando GNSS e Interface com Mapas *on-line*



**Pedro Miguel Ribeiro
Bouça Nova**

**Localização e Navegação Global do ATLASCAR2
usando GNSS e Interface com Mapas *on-line***

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Mecânica, realizada sob orientação científica de Vítor Manuel Ferreira dos Santos, Professor Associado com Agregação do Departamento de Engenharia Mecânica da Universidade de Aveiro e de Miguel Armando Riem de Oliveira, Professor Auxiliar do Departamento de Engenharia Mecânica da Universidade de Aveiro.

O júri / The jury

Presidente / President

Prof. Doutor José Paulo Oliveira Santos

Professor Auxiliar da Universidade de Aveiro

Vogais / Committee

Doutor Filipe Neves dos Santos

Investigador INESC-TEC - Porto (arguente)

Prof. Doutor Vítor Manuel Ferreira dos Santos

Professor Associado com Agregação da Universidade de Aveiro (orientador)

**Agradecimentos /
Acknowledgements**

Aos meus pais e irmãs.

À Catarina pela ajuda e paciência.

Ao Professor Vítor Santos pelo rigor e exigência colocados na orientação deste trabalho.

Ao meu colega e amigo Ricardo, pela ajuda e colaboração em vários trabalhos ao longo do curso.

A todos, muito obrigado!

Palavras-chave

ATLASCAR 2; Navegação; GNSS; INS; ROS; *Google Maps API Directions*

Resumo

Os veículos totalmente autónomos, ou veículos sem condutor, são veículos que apenas necessitam que o utilizador ou algum sistema indique o local de destino. Nesse sentido, a presente dissertação tem como objetivos obter a localização global do veículo ATLASCAR 2 e estabelecer um sistema de planeamento e execução de rotas entre a posição corrente e uma determinada posição de destino. A estas finalidades são associadas as tarefas de desenvolver uma interface gráfica em ambiente ROS e implementar um sistema de monitorização remota do estado do veículo através de uma página *web*.

A tarefa de localizar no globo o veículo foi realizada através da instalação e configuração do sensor Novatel SPAN IGM A1. Este sensor integra o sistema de navegação global por satélite com um sistema de navegação inercial, de modo a obter-se leituras de posição absoluta com grande precisão.

A navegação global utiliza os pontos de rota obtidos através da decodificação das rotas geradas pelo serviço externo *Google Maps API Directions* que, posteriormente, são geridos através de uma metodologia de planeamento global que visa fornecer ao módulo planeador local um segmento de caminho indicativo da direção que o veículo deve seguir para chegar ao destino. O planeador global está preparado para proceder ao cálculo de uma nova rota quando o veículo sai da rota estabelecida.

Todas as variáveis relativas à localização e navegação global podem ser visualizadas em tempo real através da aplicação *Mapviz*, para a qual foram desenvolvidos *plugins* que permitem ao utilizador estabelecer uma missão e acompanhar a execução da mesma.

O sistema de monitorização remota é centrado no desenvolvimento de uma base de dados em *Postgresql*, que é acedida remotamente a partir do veículo para o fornecimento de informação sobre o seu estado.

Por fim, foi testado o desempenho do sensor em situações críticas e avaliado o funcionamento da metodologia de planeamento e execução de rota desenvolvida nesta dissertação.

Nos testes realizados, o sensor mostra que mesmo em situações mais complicadas, consegue fornecer uma solução de posicionamento que não coloca em causa o funcionamento do módulo de navegação global e em condições favoráveis a incerteza na localização é bastante inferior a um metro.

Keywords

ATLASCAR 2; Navigation; GNSS; INS; ROS; *Google Maps API Directions*

Abstract

Fully autonomous vehicles, or vehicles without drivers, are vehicles that only require the user or some system to provide the destination. Therefore, this dissertation aims to obtain the global location of the ATLASCAR 2 vehicle and establish a system for planning and executing routes between the current position and a certain destination position. These purposes are related to the tasks of developing a graphical interface in ROS environment and of implementing a remote monitoring system of the vehicle's status through a web page.

The task of locating the vehicle on the globe was accomplished by installing and configuring the Novatel SPAN IGM A1 sensor. This sensor integrates a global navigation satellite system and an inertial navigation system, in order to obtain absolute position readings with great precision.

The global navigation uses the route points obtained by decoding routes generated by the Google Maps API Directions service, which are then managed through a global planning methodology that aims to provide the local planner module with a route segment indicative of the direction which the vehicle must follow to reach the destination. The global planner is prepared to calculate a new route when the vehicle leaves the established route.

All variables related to the global location and navigation can be viewed in real time through the Mapviz application, for which plugins have been developed that allow the user to establish a mission and follow the execution of the same.

The remote monitoring system is centered on the development of a database in Postgresql, which is accessed remotely from the vehicle to provide information about its status.

Finally, the sensor performance was tested in demanding situations and the operation of the route planning and execution methodology developed in this dissertation was evaluated.

In the tests carried out, the sensor shows that, even in more complicated situations, it can provide a positioning solution that does not jeopardize the operation of the global navigation module and in favorable conditions the location uncertainty is well below one meter.

Conteúdo

1	Introdução	1
1.1	Enquadramento e motivação	1
1.2	Descrição do problema	2
1.3	Objetivos	3
1.4	Trabalhos relacionados	3
1.4.1	Projeto ATLASCAR	4
1.4.2	Localização global em veículos terrestres	4
1.4.3	Sistemas de navegação por satélite	5
1.5	Estrutura da dissertação	7
2	Localização Global	9
2.1	<i>Global Navigation Satellite System</i> (GNSS)	9
2.2	Integração dos sistemas GNSS com <i>Inertial Navigation System</i> (INS) . .	11
2.3	Hodometria	14
2.4	<i>Real Time Kinematic</i> (RTK)	15
2.5	<i>Precise Point Positioning</i> (PPP)	16
3	Infraestrutura Experimental	19
3.1	NovAtel SPAN IGM A1	19
3.1.1	Instalação do sensor e da antena	21
3.1.2	Configuração do sensor	24
3.2	<i>Google Maps API Directions</i>	26
3.3	Monitorização	28
4	Arquitetura de Software	31
4.1	<i>Robot Operating System</i> (ROS)	31
4.2	Comunicação com o sensor	32
4.3	Planeamento de rota	35
4.4	Aplicação Mapviz	39
4.5	Postgresql	43
4.5.1	Base de dados local	43
4.5.2	Base de dados remota	44
4.6	Página web	45

5	Testes e resultados	47
5.1	Calibração do sensor	47
5.1.1	Campo aberto	47
5.1.2	Ambiente urbano	49
5.2	Situações propícias a falhas do sistema	52
5.2.1	Rua delimitada por edifícios	52
5.2.2	Túnel	54
5.2.3	Rua com árvores de grande porte	59
5.3	Planeamento e execução de rota	62
5.3.1	Aplicação do código da estrada	64
5.3.2	Saída de rota	65
6	Conclusões e Trabalho Futuro	69
6.1	Conclusões	69
6.2	Trabalho futuro	70
	Apêndices	73
A	Suporte	75
B	Google Maps API Directions	77
B.1	Exemplo de mensagens	77
B.2	Função de descodificação da <i>polyline</i>	79
C	Comunicação com o sensor	81
C.1	Implementação da mensagem <i>bestgnsspos</i>	81
C.2	<i>Launch file</i>	84
D	Planeamento	85
E	Mapviz	87
E.1	<i>Plugins</i> para representação da posição	87
E.2	<i>Plugins</i> para representação da rota	89
E.3	Sistema de eixos do veículo	91

Lista de Tabelas

2.1	Sistemas GNSS	11
3.1	Características técnicas do sensor <i>NovAtel SPAN IGM A1</i>	20
3.2	Características técnicas da antena <i>GPS-702-GG</i>	21
3.3	Comandos de configuração.	26
4.1	Tabela <i>destination_table</i> para registo dos locais de destino.	43
4.2	Tabela <i>waypoint_table</i> para registo dos waypoints descodificados.	43
4.3	Tabela <i>steps waypoint_table</i> para o registo dos locais de início e fim de etapa e a sua instrução associada.	44
4.4	Tabela <i>global_navigation</i> para o registo da informação relacionada com o posicionamento e navegação global.	44

Lista de Figuras

1.1	ATLASCAR 1	2
1.2	ATLASCAR 2	2
1.3	Esquema representativo de uma situação de saída de rota (Katayama et al., 2002).	6
1.4	Esquema de detecção de saída de rota (Min, 2006).	6
2.1	Componentes de um sistemas de navegação global por satélite.	9
2.2	GNSS + INS	12
2.3	Diagrama de funcionamento da ligação fraca.	13
2.4	Diagrama de funcionamento da ligação forte.	13
2.5	Ligação forte vs ligação fraca.	14
2.6	Valores de erro de posição 2D com a aplicação de um <i>encoder</i>	15
2.7	Diagrama de funcionamento da metodologia RTK.	16
2.8	Diagrama de funcionamento da metodologia PPP.	17
3.1	Sistema GNSS+INS	19
3.2	Exemplo de instalação e representação da distância entre o sensor e a antena segundo os três eixos.	22
3.3	Orientação de defeito do sensor.	22
3.4	Principais etapas da instalação do sistema constituído pelo sensor GNSS+INS e antena.	23
3.5	Resultado final da instalação do sistema constituído pelo sensor e antena.	24
3.6	Esquema de instalação, a verde o sistema de eixos do veículo e a vermelho o sistema de eixos do sensor.	25
3.7	Serviço <i>on-line Google Maps API Directions</i>	28
3.8	Estado inicial da página <i>web</i>	29
4.1	Gráfico de distribuição de tópicos para o nodo <i>novatel_position</i>	34
4.2	Gráfico de distribuição de tópicos para o nodo <i>novatel_imu</i>	34
4.3	Fluxograma de funcionamento do serviço <i>Robot Operating System (ROS)</i> que procede ao pedido de rota junto da aplicação <i>Google Maps API Directions</i>	35
4.4	Funcionamento da metodologia de planeamento de um segmento de caminho, numa situação genérica.	36
4.5	Representação gráfica do momento em que ocorre a passagem para o segmento S_3	37
4.6	Fluxograma de funcionamento do planeador global.	38

4.7	Arquitetura da aplicação <i>Mapviz</i> . <i>A</i> - Escolha do referencial do veículo; <i>B</i> - <i>Plugins</i> em utilização; <i>C</i> - Representação gráfica dos <i>plugins</i> em utilização.	40
4.8	Representação na aplicação <i>Mapviz</i> da posição a vermelho utilizando a integração GNSS+INS e a azul a posição calculada através do sistema GNSS.	40
4.9	Exemplo do funcionamento do <i>plugin global_planning</i> , representando a preto os segmentos de retas definidos pelos <i>waypoints</i> , a verde as etapas do percurso e a amarelo os pontos que definem o segmento de referência para o veículo.	41
4.10	Imagem da aplicação <i>Mapviz</i> durante o planeamento e execução de uma rota.	42
4.11	Arquitetura de funcionamento do <i>package database</i> .	45
4.12	Esquema de implementação da monitorização remota do ATLASCAR 2.	45
4.13	Página <i>web</i> da monitorização remota do ATLASCAR 2.	46
5.1	Número de satélites ao longo do período de alinhamento em condições favoráveis.	48
5.2	Valor de erro das leituras fornecidas pelo sensor durante o processo de calibração em condições favoráveis.	48
5.3	Percurso percorrido durante o processo de alinhamento com início no ponto <i>A</i> e fim no local onde o veículo se encontra.	49
5.4	Rua Prof Manuel Est. Silva em Aradas, Aveiro	49
5.5	Leituras de posicionamento global durante o processo de alinhamento com início no ponto de coordenadas (-8.643189,40.625362).	50
5.6	Número de satélites ao longo do período de alinhamento em rua delimitada por edifícios.	50
5.8	Percurso efetuado durante o processo de alinhamento com início em <i>A</i> .	51
5.9	Representação da posição global durante as três passagens na Rua Prof. Manuel Est. Silva.	52
5.10	Número de satélites utilizados durante o período de tempo considerado na figura 5.9.	53
5.11	Valores de erro2D durante as passagens da rua Prf. Manuel Est. Silva.	54
5.12	Perfis dos túneis utilizados para a realização dos testes.	55
5.13	Repetibilidade das leituras de posicionamento global na passagem pelo túnel na Av. Santa Joana.	55
5.14	Representação do número de satélites utilizados na solução e velocidade do veículo durante o período correspondente ao deslocamento representado na figura 5.13.	56
5.15	Valores de erro da posição global, calculados segundo a equação 5.1.	57
5.16	Representação no mapa, em vista satélite, das leituras de posicionamento global durante a passagem pelo túnel da Av. Lourenço Peixinho.	57
5.17	Representação no mapa, em vista satélite, da posição global utilizando GNSS+INS (a vermelho) e da posição global apenas com base no sistema GNSS (a azul).	58
5.18	Repetibilidade das leituras de posicionamento global nas quatro passagens pela Av. Artur Ravara.	59

5.19	A colorido número de satélites utilizados pelo sensor no cálculo da posição e a preto o valor de velocidade do veículo nas quatro passagens pela Av Artur Ravara.	60
5.20	Valor de erro utilizando a equação 5.1 durante a passagem pela Av. Artur Ravara.	61
5.21	Representação no mapa em vista satélite da posição global utilizando GNSS+INS (a vermelho) e da posição global apenas com base no sistema GNSS (a azul)	62
5.22	Rota entre A e B usada no estudo do funcionamento do algoritmo de planeamento desenvolvido.	63
5.23	Valor de S_i (a vermelho) e distância ao <i>waypoint</i> seguinte (a azul) em função do tempo.	63
5.24	Demonstração do planeamento numa situação em que o destino (B) da rota se encontra numa via de sentido único.	65
5.25	Execução de rota entre o ponto A e B	66
5.26	Recalculo da rota entre o ponto C e B depois de detetada a saída de rota.	66
5.27	Execução de rota entre o ponto A e B	67
5.28	Recalculo da rota entre o ponto C e B depois de detetada a saída de rota.	67
6.1	Situação e posicionamento errado dos <i>waypoints</i>	70

Lista de Acrónimos

DEM Departamento de Engenharia Mecânica. 1, 4

GEO *Geostationary Earth Orbit* . 11

GNSS *Global Navigation Satellite System*. 3

GPS *Global Positioning System*. 9

IGSO *Inclined Geo-Synchronous Orbit* . 11

INS *Inertial Navigation System*. i, 11

LAR Laboratório de Automação e Robótica. 3, 4, 28, 43, 44

MEO *Medium Earth Orbit* . 11

OSRM *Open Source Routing Machine*. 4

PPP *Precise Point Positioning*. 9

ROS *Robot Operating System*. v, 3, 4, 29, 31, 32, 35, 39, 41, 43, 44

RTK *Real Time Kinematic*. 9

SPAN *Synchronized Position, Attitude and Navigation*. 19

Capítulo 1

Introdução

1.1 Enquadramento e motivação

Os veículos totalmente autónomos, ou veículos sem condutor, são veículos que apenas necessitam que o utilizador forneça o local de destino. Os avanços tecnológicos recentes na área dos sistemas avançados de assistência à condução, permitiram a construção de veículos reais com capacidade suficiente para serem testados em ambiente real. A complexidade da navegação autónoma levou ao desenvolvimento de sistemas de apoio à condução e sistemas de segurança para maior proteção dos agentes humanos. Deste modo, é necessário instrumentar os veículos com diversos sensores como, por exemplo, *lasers*, câmaras e sensores de posicionamento global. Esta dissertação detalha a instalação de um sensor de posicionamento global num veículo instrumentado para obtenção da localização do veículo no mundo em tempo real. Esta localização permitirá encontrar uma rota entre a posição do veículo e o local de destino fornecido pelo utilizador recorrendo a serviços *on-line*.

O projeto ATLAS, criado em 2003 pelo grupo de Automação e Robótica do Departamento de Engenharia Mecânica (DEM) da Universidade de Aveiro, dedica-se atualmente ao desenvolvimento de sistemas de apoio à condução. Inicialmente foram desenvolvidos protótipos de pequena escala e, em 2010, o projeto evoluiu para um veículo à escala real, o ATLASCAR 1, figura 1.1. Em 2016 o projeto migrou para um Mitsubishi i-Miev - ATLASCAR 2 - um veículo totalmente elétrico que permite diminuir o número de atuadores face ao ATLASCAR 1. Nesta fase, o ATLASCAR 2, figura 1.2, encontra-se equipado com dois *laser scanner* (LIDAR) 2D e um 3D, uma câmara e um computador com elevada capacidade de processamento (Correia, 2017).

Esta dissertação estabelece uma variedade de tarefas com características muito diferentes, criando desta forma um grande desafio à sua execução. A solução de navegação global a desenvolver incluí localizar com rigor o veículo, utilizando um sensor moderno com características técnicas ideais à aplicação em causa, e fornecer ao módulo de navegação local uma rota, proveniente de um serviço *on-line*, para que o veículo consiga chegar ao destino introduzido pelo utilizador. Aliado a isto, é necessário estabelecer um sistema de monitorização capaz de apresentar informação proveniente do sensor de localização e outros parâmetros considerados relevantes numa página *web*.



Figura 1.1: ATLASCAR1 Ford Escort SW.



Figura 1.2: ATLASCAR 2 e localização dos principais sensores.

1.2 Descrição do problema

Navegação é a ciência (ou arte) de conduzir um robô móvel a atravessar um determinado ambiente (por terra, mar ou ar). Inerente ao processo de navegação está o desejo de chegar a um destino sem perder o caminho (rota) e sem colisões. A navegação engloba três tarefas: mapeamento, planeamento e condução (McKerrow, 1991).

Historicamente, o ato de navegar efetuava-se apenas recorrendo a referências visuais, algo que os seres humanos ainda hoje fazem; contudo, a navegação marítima fomentou o desenvolvimento de novas metodologias e, atualmente, as técnicas de navegação global disponíveis permitem responder de forma eficaz a muitos desafios.

Da mesma forma que a navegação marítima impulsionou o desenvolvimento das técnicas de navegação, o desafio de criar veículos autónomos veio impor o desenvolvimento de técnicas de geolocalização de elevada precisão, o que a tecnologia *Global Navigation Satellite System* (GNSS) em determinadas situações não consegue fornecer. Em ambientes com elevadas restrições de visibilidade, como é o caso dos ambientes urbanos, as reflexões de sinal podem ocasionar erros de mais de 50 metros (Wu et al., 2017). No limite, existe a perda de sinal do número mínimo requerido de satélites (4) para o cálculo da posição, o que em aplicações de veículos autónomos facilmente acontece na passagem de um túnel ou no simples ato de estacionamento num parque subterrâneo. É devido a estes fatores que atualmente a tecnologia GNSS necessita obrigatoriamente de ser complementada.

Aplicações com veículos autónomos requerem uma precisão de localização global inferior a 1 metro (Roysdon et al., 2017).

A tarefa da navegação global é encontrar um caminho entre dois pontos, contudo, é necessário que esse caminho cumpra determinados requisitos, como a aplicação das regras do código da estrada e, numa perspetiva geral, o módulo que efetua o planeamento global deve ser dependente de um serviço com uma grande cadência de atualização. Deste modo, é possível planear uma rota com base nos traçados de estrada mais recentes. Imagine-se a confusão que seria para o veículo quando este estivesse à espera de nas proximidades encontrar um entroncamento e encontrasse uma rotunda.

A geolocalização é fundamental para o sucesso dos veículos inteligentes, quer seja veículos totalmente autónomos ou em aplicações de gestão de frotas. A sua localização deve ser conhecida de forma a que aplicações como execução de trajetórias entre dois pontos do globo ou mesmo a escolha de trajetórias condicionadas pelo volume de tráfego possam funcionar acertadamente.

1.3 Objetivos

Os objetivos pretendidos com a presente dissertação são os seguintes:

- Instalação e configuração do recetor GNSS +INS - Novatel SPAN IGM A1 - no veículo.
- Desenvolvimento de uma aplicação integrada em ROS para representar a posição absoluta do veículo.
- Desenvolvimento de um sistema de planeamento de rotas para navegação global.
- Desenvolvimento de um sistema de monitorização constituído por uma base de dados capaz de guardar informação relevante durante as missões do veículo.
- Representação da posição do veículo, em tempo real, numa página web em ambiente *Google Maps*.

1.4 Trabalhos relacionados

Nesta secção é feito o levantamento dos trabalhos realizados no Laboratório de Automação e Robótica (LAR) que constituem as linhas orientadoras para a realização das tarefas

descritas na secção 1.3. São também considerados outros trabalhos com características semelhantes a esta dissertação.

1.4.1 Projeto ATLASCAR

O trabalho de Ferreira, (2015), presente do repositório do LAR do DEM da Universidade de Aveiro, estabeleceu o primeiro contacto com sensor GNSS+INS atualmente disponível para instalação no ATLASCAR 2. Este sensor foi instalado no ATLASCAR1 e integrou uma solução de navegação global que a partir da localização proveniente do sensor GNSS e, de um ponto de destino, conseguia planejar uma rota e fornecer indicações para a executar; a informação relativa às rotas é proveniente da aplicação *Open Source Routing Machine* (OSRM). Foi também, desenvolvida/adaptada uma interface web para representação da localização do veículo e uma base de dados para arquivo de informação recolhida durante os testes.

A presente dissertação estabelece o trabalho de Ferreira, (2015), como principal referência, e dele será extraída a visão geral do trabalho. Contudo, será percorrido um caminho diferente uma vez que a instalação do sensor e antena será efetuada de acordo com as especificações presentes no seu manual. No campo do planeamento de rotas será utilizada a aplicação *Google Maps API Directions* e a interface com o utilizador será substituída por uma aplicação em ROS.

Também no repositório do LAR, o trabalho de Viana, (2013) encontra-se detalhado o desenvolvimento de uma solução de controlo remoto do ATLASCAR1 e a criação da atual página *web* disponível. Neste trabalho encontra-se documentado todo o processo de criação da página referida, desde a criação da base de dados em *Postgresql* localizada no servidor do LAR, elemento central no sistema de monitorização, passando pelo ficheiro em linguagem *javascript*, responsável pela atualização do *widget* do *Google Maps*, e todos os outros campos com variáveis provenientes do veículo com uma frequência de 5Hz e pelo ficheiro em *php* que tem a seu cargo a tarefa de ler a base de dados e fornecer as leituras ao ficheiro *javascript*.

Neste caso, o sistema de monitorização não se encontra funcional, o nodo ROS responsável pela interface com a base de dados em *Postgresql* está obsoleto, pelo que requer a criação e um nodo totalmente novo, e adequação da página *web* às variáveis disponíveis do ATLASCAR 2.

1.4.2 Localização global em veículos terrestres

A localização global de veículos autónomos requer leituras com elevada precisão (melhor do que 1 metro), o que os sensores comuns de navegação global por satélite não conseguem fornecer. Neste sentido, o sensor utilizado neste trabalho integra um recetor GPS e um sistema de navegação inercial (INS) capaz de fornecer leituras, em determinados ambientes, com um intervalo de erro inferior a 1 metro. O trabalho realizado por Houghton et al., (2013) consiste em averiguar a magnitude de possíveis erros em ambientes urbanos propícios à ocorrência de fenómenos de reflexão e estudar possíveis soluções como a instalação de um hodómetro. Este trabalho será utilizado como base de comparação nos testes desta dissertação relativos à resposta do sensor em diversos ambientes.

No campo do planeamento de rota o trabalho desenvolvido por Lemos et al., (2015), permite comparar o desempenho de dois serviços *on-line* capazes de fornecer uma rota

entre dois locais. Este trabalho desenvolve ainda uma metodologia capaz de traçar uma curva entre sucessivos pontos de passagem que não será objeto de estudo nesta dissertação.

Segundo Paden et al., (2016), uma rota entre dois pontos é um conjunto de segmentos de caminho que definem a direção a seguir pelo veículo em cada instante. Este trabalho pretende estabelecer uma metodologia para encontrar o segmento de caminho que o veículo deve utilizar para chegar ao seu destino.

1.4.3 Sistemas de navegação por satélite

Atualmente estão disponíveis diversos tipos de equipamentos de navegação por satélite, desde *apps* para *smartphone* até dispositivos integrados em automóveis. Estes sistemas têm como objetivo fornecer indicações de navegação para que o utilizador consiga deslocar-se de um local A para um local B.

Os dispositivos podem ter um conjunto de opções relativas ao tipo de veículo, critério de escolha de rota como, por exemplo, a distância percorrida ou o tempo previsto de execução da rota (ZENEC, 2013).

Estes equipamentos procedem ao planeamento de uma rota e verificam a sua execução. No contexto desta dissertação, o procedimento de verificação da execução de rota e deteção de uma possível saída de rota será utilizado como referência. Em 2002, a Honda Motor patenteou um sistema de navegação por satélite que procede ao planeamento de rota e verifica se o veículo saiu de rota. O equipamento deteta a saída de rota caso a distância P_{min} for superior a um determinado valor (figura 1.3). A título de exemplo são referidos 50 metros como distância predefinida para a saída de rota (Katayama et al., 2002).

Depois de ser detetada a saída de rota é criado um novo troço na rota (*return line*), que visa guiar o veículo da posição atual à posição mais próxima da rota planeada.

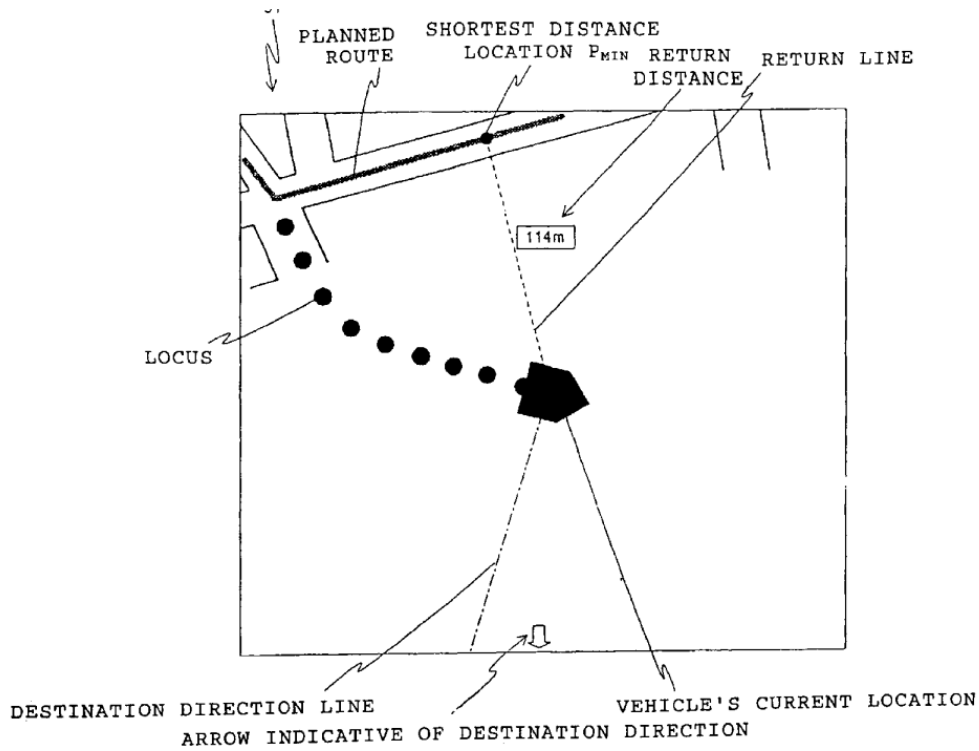


Figura 1.3: Esquema representativo de uma situação de saída de rota (Katayama et al., 2002).

Em 2006, a Samsung Electronics patenteou um sistema de navegação por satélite em que a saída de rota é encontrada através de uma distância limite (D), entre a posição do veículo (P_b) e os pontos de rota P_a e P (figura 1.4), sempre que a distância D é superior a um determinado valor é realizado o cálculo de uma nova rota (Min, 2006).

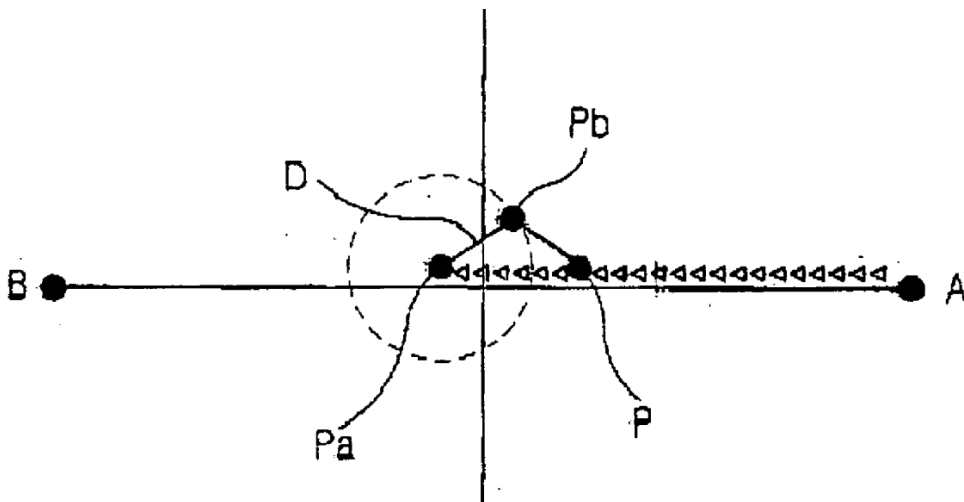


Figura 1.4: Esquema de detecção de saída de rota (Min, 2006).

1.5 Estrutura da dissertação

A presente dissertação é constituída por seis capítulos, organizados da seguinte forma:

1. **Introdução** - Neste capítulo é feito um breve enquadramento da problemática em estudo e são descritos alguns trabalhos relacionados.
2. **Localização Global** - Estudo dos sistemas GNSS e sistemas complementares capazes de aumentar a precisão das leituras obtidas em aplicações de veículos autónomos.
3. **Infraestrutura Experimental** - Apresentação do *hardware* e ferramentas utilizadas no desenvolvimento do sistema geral.
4. **Arquitetura de Software** - Descrição detalhada da arquitetura do software aplicado no funcionamento do sistema.
5. **Testes de Resultados** - Demonstração e análise do funcionamento do sistema.
6. **Conclusão** - Apresentação das conclusões do trabalho e proposta de trabalhos futuros.

Capítulo 2

Localização Global

Este capítulo descreve os conceitos básicos inerentes ao funcionamento do sistema de navegação global por satélite e apresenta algumas metodologias para aumentar o nível da precisão das leituras de geolocalização, nomeadamente, a fusão com um sistema de navegação inercial, a utilização de um hodómetro e as técnicas *Real Time Kinematic* (RTK) e *Precise Point Positioning* (PPP).

2.1 *Global Navigation Satellite System* (GNSS)

No final da década de 70 o Departamento de Defesa dos Estados Unidos lançou o *Global Positioning System* (GPS); desenvolvido para aplicações militares, foi o primeiro sistema global de navegação por satélite (GNSS) e é considerado o maior avanço nas técnicas de navegação até ao momento (NovAtel, 2015a).

O funcionamento dos sistemas de navegação por satélite pode ser dividido em 5 partes principais como pode ser observado na figura 2.1.

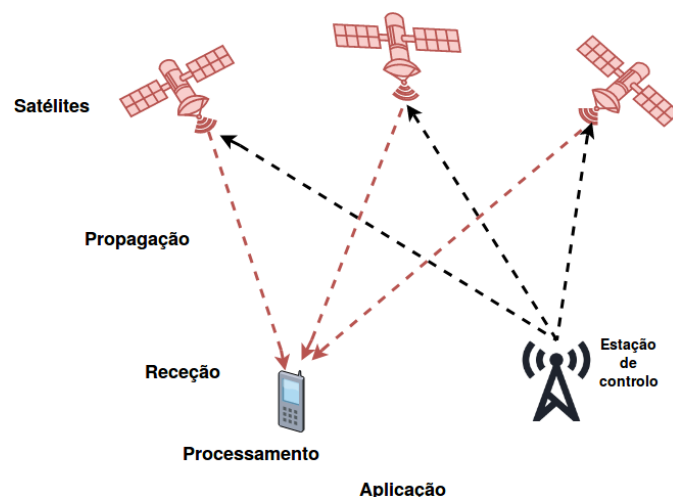


Figura 2.1: Componentes de um sistemas de navegação global por satélite.

As cinco componentes, ou etapas do processo são as seguintes:

1. Constelação de satélites que orbitam em torno do planeta a uma altitude média de 20 000 km relativamente à superfície da terra, pesam aproximadamente 1 500 kg, são equipados com relógios extremamente precisos e são controlados através de estações de controlo espalhadas pelo planeta.
2. Propagação de sinais provenientes de satélites para os utilizadores e sinais gerados nas estações de controlo com destino aos satélites. Estes sinais de rádio atravessam várias camadas da atmosfera onde podem ocorrer fenómenos de refração, a propagação ao longo da ionosfera pode provocar erros na estimativa da posição de 5 metros (NovAtel, 2015a). O ambiente urbano, com elevados constrangimentos à propagação de sinais, é umas das principais fontes de erro dos sistemas de navegação por satélite.
3. O equipamento do utilizador recebe sinais de diferentes satélites e determina o tempo entre o envio do sinal e a receção do sinal através da descodificação da mensagem de navegação ou, em equipamentos com maior capacidade, é utilizada a onda portadora para a determinação do número de ciclos entre a emissão e receção. Sabendo que as ondas portadoras têm um comprimento de onda constante, multiplicando esse valor pelo número de ciclos determina-se a distância entre o recetor e o satélite. Esta etapa é denominada por Receção.
4. Processamento da informação recolhida na etapa anterior e através do método de trilateração utilizando a distância entre o utilizador e três satélites é possível estimar a posição do utilizador - Processamento.
5. Uso da posição numa determinada aplicação, por exemplo navegação aérea, terrestre ou marítima, mapeamento, construção civil, agricultura.

Os sistemas de localização por satélite têm sofrido alterações desde o seu lançamento. Na tabela 2.1 são apresentadas algumas das características dos principais sistemas em funcionamento.

Como é possível observar na tabela 2.1, algumas constelações ainda não estão em total funcionamento, o que significa que irá ocorrer nos próximos tempos uma evolução significativa na qualidade dos serviços oferecidos pelos sistemas GNSS. A modernização do sistema GPS, que em breve lançará o quarto sinal de satélite civil (L1C) compatível com sistema Galileo, assegura aos utilizadores a convergência dos recursos de modo a garantir o aumento de precisão das leituras fornecidas (NovAtel, 2015a).

O funcionamento dos sistemas de localização por satélite apresenta vários fatores que dificultam o cálculo da posição exata do recetor, entre os quais se destacam, erros na compensação dos relógios dos satélites, erros derivados das variações de órbita, refração dos sinais durante a propagação ao longo da ionosfera e em ambiente com elevados constrangimentos é comum a ocorrência de fenómenos de reflexão. A calibração dos relógios dos satélites com um erro de 10 nanossegundos pode provocar um erro de posição de 3 metros (NovAtel, 2015a). A ionosfera, camada da atmosfera situada a uma altitude entre os 80 km a 600 km da superfície da terra, contém partículas com carga elétrica capaz de interferir na normal propagação dos sinais provenientes dos satélites. Em ambientes urbanos com edifícios com várias centenas de metros podem levar à ocorrência de reflexão da uma onda várias vezes antes da sua receção por parte do recetor, em casos limites, podem surgir zonas em que o serviço GNSS não está disponível.

Tabela 2.1: Características dos sistemas GNSS (Teunissen et al., 2017).

	GPS	GLONASS	GALILEO	BeiDou
Satélites	27 + 4	24 + 3	27 +3	35
Órbita	MEO	MEO	MEO	*
Planos orbitais	6	3	3	3
Inc. da órbita	55°	64.8°	56°	0°+55°+55°
Altitude	20 200 km	19 140 km	23 222 km	21528 a 35787 km
Frequências	L1, L2, L5	L1, L2	E1, E5, E6	B1, B2, B3
Período	11 h 58 m	11h 15m 44s ±5s	14h 04m 42 s	12h 53m
Serviços	SPS, PPS	SPS, PPS	OS, CS, PRS	OS, AS, WADS, SMS
Funcionamento	total	total	2020 esperado	2020 esperado

* 5 *Geostationary Earth Orbit* (GEO) +3 *Inclined Geo-Synchronous Orbit* (IGSO) +27 *Medium Earth Orbit* (MEO)

SPS - *Standart Positing Service*; PPS - *Precise Positing Service*; OS - *Open Service*; CS - *Commercial Service*; PRS - *Public Regulated Service*; AS - *Authorized Service*; WADS - *Wide Area Differential Service*; SMS - *Short Message Service*;

Nas secções seguintes são apresentadas três técnicas para reduzir os erros descritos no parágrafo anterior.

2.2 Integração dos sistemas GNSS com INS

A precisão da leitura de geolocalização pode ser aumentada integrando uma unidade inercial ao sistema GNSS (Won et al., 2014). Nesta secção será apresentada a integração do recetor GNSS com um sistema de navegação inercial (INS).

Como referido na secção 2.1, os sistemas de navegação global por satélite usam sinais provenientes dos satélites em órbita para calcular a posição do utilizador. Devido a problemas de reflexão, frequentes em ambientes urbanos, podem ocorrer erros nas leituras de várias dezenas de metros. Assim sendo, a forma encontrada para minimizar estes erros em aplicações de navegação autónoma foi a integração da informação proveniente do recetor GNSS com parâmetros cinemáticos. Segundo Cui et al., (2017), a integração entre recetores GNSS e um sistema de navegação inercial foi amplamente aprovada como substituta do recetor GNSS autónomo na navegação em áreas urbanas.

A figura 2.2 mostra o exemplo de uma situação de grande reflexão dos sinais devido à presença de edifícios de elevadas dimensões, em que o sistema de navegação inercial assegura o correto posicionamento durante alguns segundos.

O sistema de navegação inercial usa as medições discretas de rotação e aceleração provenientes da unidade inercial para calcular uma posição relativa ao longo do tempo. A unidade inercial é constituída por 3 acelerómetros e 3 giroscópios distribuídos ao longo de um sistema de eixos ortogonais. Os acelerómetros medem as acelerações lineares enquanto os giroscópios medem as velocidades angulares, dados utilizados na integração numérica em ordem ao tempo de modo a calcular uma posição relativa. É de salientar que o sistema inercial de navegação fornece continuamente uma posição e não é afetado por campos magnéticos, contudo, a solução inercial sofre uma rápida degradação de

precisão quando não tem o auxílio do sistema de navegação por satélite.

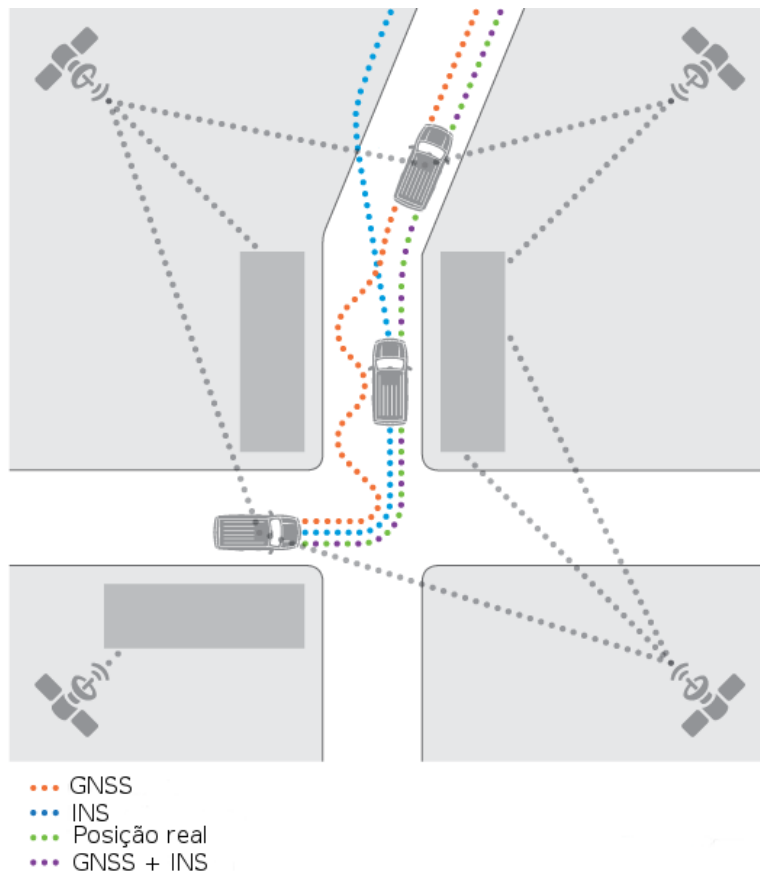


Figura 2.2: Fusão do sistema global de navegação por satélite com o um sistema de navegação inercial (adaptado de NovAtel, 2015a).

A integração do sistema de navegação por satélite com um sistema de navegação inercial é realizado através de uma ligação fraca (*loosely coupled*) ou de uma ligação forte (*tightly coupled*).

Na ligação fraca, o recetor GNSS usa os sinais recebidos, provenientes da constelação de satélites, para calcular uma posição e velocidade da antena GNSS. O sistema INS utiliza as medições de aceleração e rotação para estimar a posição e velocidade da antena. Um filtro de Kalman estendido usa os valores de posição e velocidade residuais, assim como modelos de erros para estimar o valor de erro geral resultante da fusão dos dois sistemas e desta forma eliminá-lo (figura 2.3).

Na ligação forte, apesar de ser usado também um filtro de Kalman estendido, a integração dos dois sistemas é realizada a um nível superior; são fundidas as distâncias entre o recetor e os satélites (figura 2.4). O sistema inercial utiliza a posição e velocidade relativa juntamente com as equações que descrevem as órbitas dos satélites para estimarem a distância entre o recetor e os satélites. Nesta metodologia, qualquer informação proveniente do sistema GNSS pode ser usada na solução, mesmo que estejam disponíveis menos de 4 satélites.

A ligação fraca faz a fusão dos valores de posição e velocidade provenientes dos

sistemas GNSS e INS, enquanto a ligação forte faz a fusão de valores de distância entre o recetor e os satélites provenientes tanto do sistema GNSS como do sistema inercial, de modo a obter o valores finais de posição e velocidade do veículo.

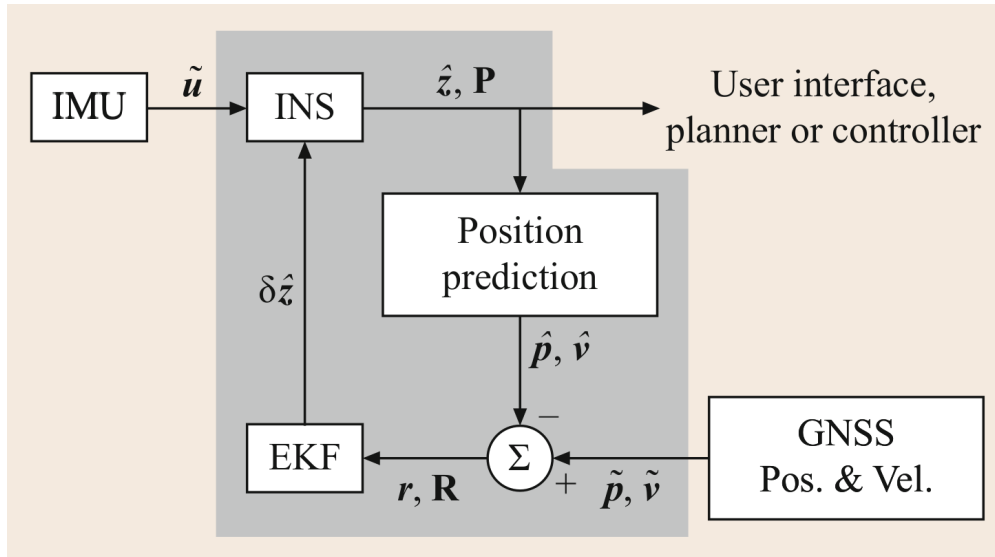


Figura 2.3: Diagrama de funcionamento da ligação fraca (Teunissen et al., 2017).

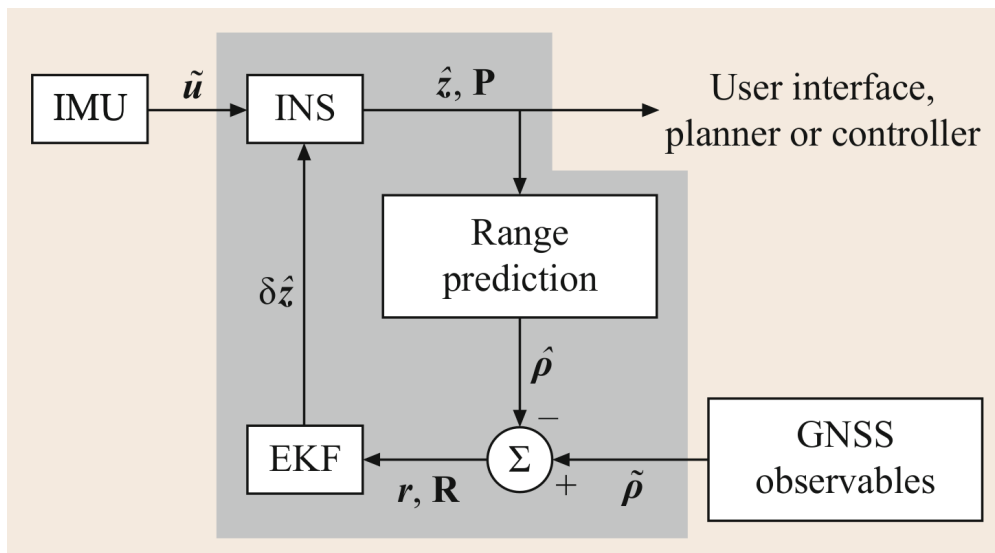


Figura 2.4: Diagrama de funcionamento da ligação forte (Teunissen et al., 2017).

A figura 2.5 mostra o resultado de um teste de comparação entre as duas integrações, onde fica clara superioridade da ligação forte face à ligação fraca. Contudo, a ligação forte exige um conhecimento mais profundo do funcionamento do sistema global de navegação por satélite (como é o caso da resolução da ambiguidade de ciclos da onda portadora) e necessita de um maior esforço computacional.



Figura 2.5: Comparação da ligação forte face à ligação fraca (Teunissen et al., 2017).

2.3 Hodometria

Um hodómetro, também conhecido por *encoder*, é um instrumento de medição de distâncias com base no número de rotações da roda do veículo. Sensores como hodómetros podem ajudar a aumentar a qualidade da solução proveniente das tecnologias GNSS+INS (NovAtel, 2015a), esta solução pode tornar-se bastante relevante em situações de observação de poucos satélites, ou mesmo na perda total de sinal.

Como demonstrado em Houghton et al., 2013, a utilização de um sensor para efetuar a medição do deslocamento independente, como é o caso dos *encoders*, em situações de visibilidade de pelo menos 4 satélites não é vantajosa, contudo em situações urbanas em que o número de satélites visíveis sejam dois ou três, pode contribuir de forma significativa para diminuição de erros. Os resultados obtidos naquele estudo mostram que em situações em que o sistema GNSS não esteja disponível, a adição de um hodómetro pode levar a diminuição de erro de quase 50% para um período de tempo de 60s sem sinal.

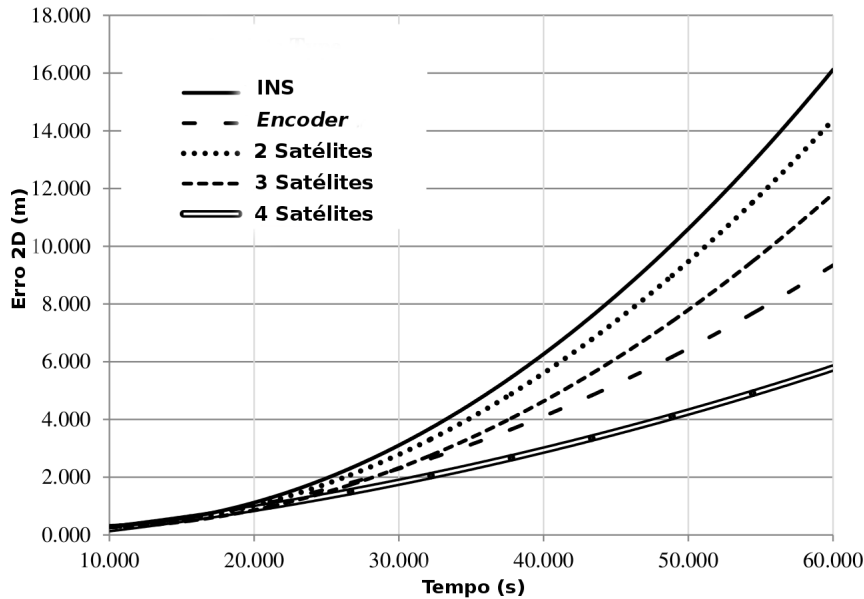


Figura 2.6: Propagação do erro de posição 2D em situações com baixo número de satélites disponíveis com a adição de um *encoder* (adaptado de Houghton et al., 2013).

A figura 2.6 permite não só avaliar o uso de um *encoder*, como possibilita analisar, de forma parcial, a interação entre o sistema de navegação por satélite e o sistema de navegação inercial, assunto tratado na secção 2.2. É de referir que o uso de uma ligação forte na fusão dos sistemas GNSS+INS é mais vantajoso do que a adição de um *encoder*, pois permite controlar com maior rigor a orientação relativamente ao norte geográfico, o que em aplicações de navegação, quer sejam terrestres, aéreas ou marítimas, é de extrema relevância.

2.4 Real Time Kinematic (RTK)

O posicionamento cinemático relativo (RTK) é uma técnica de posicionamento global que utiliza informação (diferença de fase, número de ciclos, comprimento de onda) das ondas portadoras para o cálculo da posição global. Deste modo, ocorre uma subida do nível de precisão face à técnica de calcular a distância entre o recetor e os satélites através da informação codificada na mensagem de navegação. O cálculo da distância entre o recetor e o satélite, utilizando um intervalo de tempo e a velocidade da luz, é afetada pela dessincronização dos relógios dos satélites face ao relógio do recetor, o que leva geralmente, à designação de pseudo-distância. A técnica RTK estabelece uma metodologia de posicionamento relativo ao utilizar informação proveniente dos satélites e de uma base com localização conhecida o que permite diminuir, ou em determinadas condições eliminar, os erros decorrentes da propagação do sinal ao longo da ionosfera, da órbita dos satélites e erros na sincronização dos relógios (NovAtel, 2015a).

Esta metodologia necessita de um maior número de componentes face ao sistema normal composto pelo recetor e pela antena GNSS, é necessária a instalação de uma antena e um sistema capaz de receber a informação dos satélites e enviar informação ao utilizador, a figura 2.7 mostra a configuração do sistema RTK.

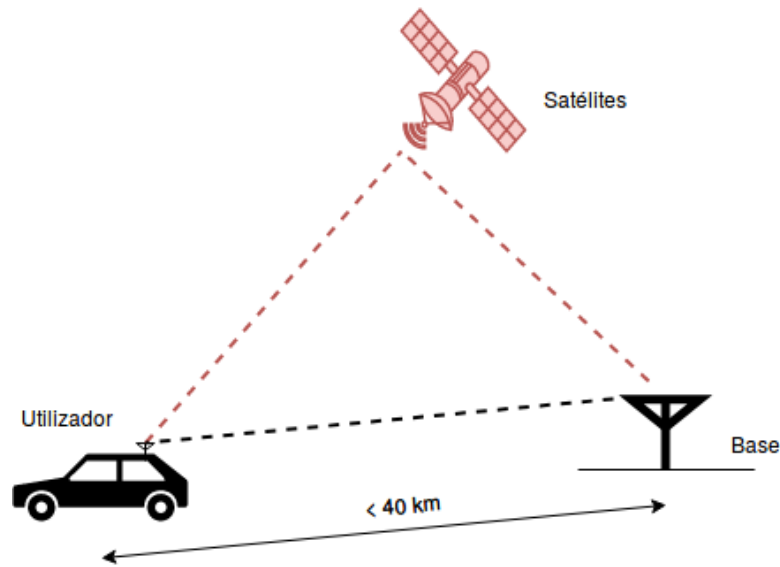


Figura 2.7: Diagrama de funcionamento da metodologia RTK.

A metodologia RTK é capaz de atingir estimativas de posição com uma gama de precisão na ordem do centímetro, sem necessitar de modelos de correção de erros relativos ao tempo dos relógios dos satélites, às órbitas descritas pelos mesmos ou modelo geofísico da Terra (NovAtel, 2015c). Contudo, a precisão das leituras é afetada pela distância entre o utilizador e a base, pelas condições de visibilidade entre a constelação e os recetores e no caso de distâncias superiores a 40 quilómetros o sistema de posicionamento relativo pode deixar de funcionar.

Em aplicações como veículos autónomos a solução RTK apresenta graves lacunas, como o preço de instalação e o funcionamento em condições ideais é realizado apenas num raio inferior a 20/30 quilómetros relativos à posição da base, o que torna imperativo o estabelecimento de uma rede de estações de base de modo a cobrir uma maior área de território, e o nível de precisão continua dependente de fenómenos locais como a reflexão.

2.5 *Precise Point Positioning (PPP)*

A técnica de posicionamento global PPP minimiza ou elimina os erros inerentes ao sistema GNSS de modo a obter um nível de precisão na ordem dos 10 centímetros com apenas um recetor (NovAtel, 2015a). A localização através da técnica PPP, figura 2.8, depende das correções à sincronização dos relógios e órbitas dos satélites geradas pelas estações de controlo e são enviadas para os utilizadores via internet ou via satélite. Esta técnica foi apresentada em 1997 e representa-se como uma alternativa muito atrativa ao sistema diferencial como é o caso da técnica RTK.

Esta metodologia requer um período de convergência superior a 30 minutos para aplicar todas as correções recebidas, no entanto em ambientes com elevados constrangimentos pode levar a um aumento significativo do tempo de convergência (NovAtel, 2015c).

A técnica PPP para determinar a ambiguidade da onda portadora utiliza um elevado número de medições de um único recetor para convergir gradualmente para a solução,

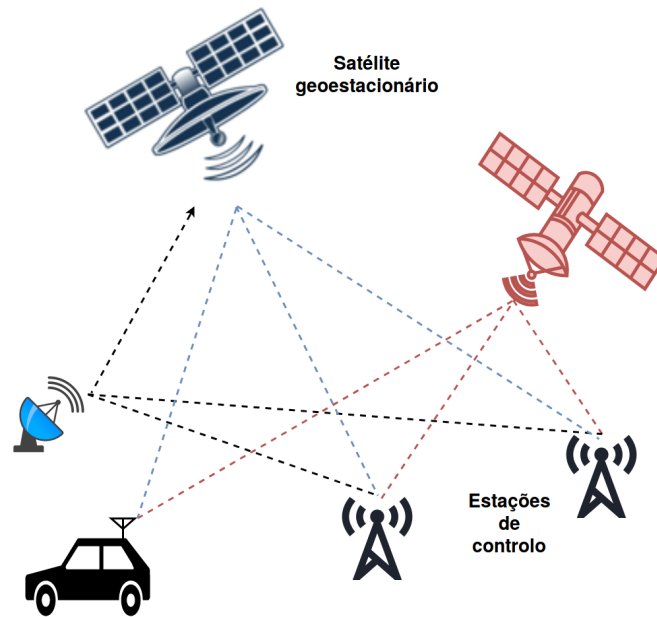


Figura 2.8: Diagrama de funcionamento da metodologia PPP.

diferindo assim da técnica RTK que explora os dados disponíveis pelos dois recetores para simplificar o problema. As duas técnicas apresentam metodologias diferentes para resolução do mesmo problema, que é a determinação do número exato de ciclos entre o envio e a receção dos sinais.

No sentido de minimizar ou eliminar as fontes de erro decorrentes do sistema GNSS, a técnica PPP opera com uma dupla frequência para minimizar o erro decorrente da propagação do sinal na ionosfera; como parte deste erro é proporcional à frequência da onda, pode ser estimado através da utilização de duas ondas com frequências diferentes. Para resolver as dessincronizações dos relógios e incorreções na órbita, utiliza dados provenientes de satélites geostacionários disponibilizados por serviços atualmente sujeitos a subscrição, como por exemplo *TerraStar* ou *OmniSTAR*. Esta informação é fundida através de um filtro de Kalman estendido.

Em aplicações como veículos autónomos a metodologia PPP apresenta numa primeira análise condições suficientes para ser aplicada, pois apenas necessita de um recetor, o que faz diminuir o preço do equipamento e evita a dependência do nível de precisão à distância relativa à base, como é o caso da técnica RTK. Contudo apresenta duas grandes desvantagens, na atualidade necessita de um elevado tempo de convergência e de um serviço externo capaz de fornecer as correções necessárias.

Capítulo 3

Infraestrutura Experimental

Este capítulo apresenta o recetor GNSS+INS e descreve o processo da sua instalação e configuração a bordo do veículo ATLASCAR 2. É apresentado o serviço *on-line Google Maps API Directions*, que permite estabelecer uma rota entre dois pontos, tendo em conta diversas variáveis, como é caso do tráfego ou a possibilidade de diferentes rotas entre os mesmos locais. Por fim, é apresentada a metodologia implementada de monitorização do veículo, tanto a nível geral (LAR), como a nível local (ATLASCAR 2).

3.1 NovAtel SPAN IGM A1

O sensor *NovAtel SPAN IGM A1*, figura 3.1a, integra a tecnologia *Synchronized Position, Attitude and Navigation* (SPAN) caracterizada pela ligação forte (*tightly coupled*) do recetor GNSS com o sistema de navegação inercial constituído pela *ADIS-16488*, uma unidade inercial de elevado desempenho. A figura 3.1b, mostra a antena GNSS *GPS-702-GG* a instalar juntamente com o sensor mencionado.



(a) Sensor *NovAtel SPAN IGM A1* (NovAtel, 2017). (b) Antena *GPS-702-GG* (NovAtel, 2018).

Figura 3.1: Sistema GNSS+INS

Na tabela 3.1 é possível observar as características técnicas do sensor, entre as quais se destacam a elevada precisão, a grande taxa de transmissão de dados relativos ao posicionamento global (até 20Hz) e às variáveis de orientação (*roll*, *pitch*, *azimuth*, até 200 Hz) e a existência de uma porta com capacidade de ser ligada diretamente a um hodómetro.

Como demonstrado na secção 2.3, a inclusão de um hodómetro no sistema constituiu, em determinadas situações, um auxílio importante para o seu bom desempenho. Contudo, não existe, neste momento, nenhum hodómetro disponível para ser instalado no veículo.

Tabela 3.1: Características técnicas do sensor *NovAtel SPAN IGM A1* (NovAtel, 2016).

Desempenho geral	
Constelação	GPS
Precisão (RMS)	
L1/L2	1.2 m
SBAS	60 cm
DGPS	40 cm
RTK	1 cm +1 ppm
Transmissão de dados	
Medição GNSS	20 Hz
Leituras GNSS	até 20 Hz
Medição IMU	200 Hz
Leituras INS	até 200 Hz
Precisão tempo	20 ns RMS
Velocidade máxima	515 m/s
Desempenho IMU	
Giroscópios	
Gama de entrada	± 450 °/s
<i>Rate bias stability</i>	6 °/h
<i>Angular random walk</i>	0.3 °/ \sqrt{h}
Acelerómetros	
Gama	± 18 g
<i>Bias stability</i>	0.1 mg
<i>Velocity random walk</i>	0.029 m/s/ \sqrt{h}
Comunicação	
1 - USB	12 Mbps
1 - RS-232 ou RS-422	921,600 bps
1 - RS-232	921,600 bps
1 CAN	1 Mbps
1 - Hodómetro	
Características físicas e elétricas	
Dimensões	152 x 142 x 51 mm
Peso	515 g
Potência	
Tensão de entrada	10-30 VDC
Consumo	4 W
Tensão antena	5 VDC $\pm 5\%$
Corrente máxima	100 mA

O centro de fase da antena *GPS-702-GG* permanece constante relativamente ao norte e o ângulo de altitude dos satélites varia (NovAtel, 2015b). Deste modo, a receção do sinal não é afetada pela orientação da antena nem pela altitude dos satélites. Na tabela 3.2, observa-se as principais características da antena disponível, entre as quais se destacam a compatibilidade com o sistema GLONASS e a receção de duas frequências (L1 e L2).

Tabela 3.2: Características técnicas da antena *GPS-702-GG* (NovAtel, 2009).

Desempenho	
Constelação	GPS , GLONASS
Frequências	
L1	1588.5±23.0 MHz
L2	1236±18.3 MHz
Banda rejeitada	
L1±100 MHz	30 dBc
L2±200 MHz	50 dBc
Ganho LNA	29 dB
Ganho <i>Roll-Off</i>	
L1	13 dB
L2	11 dB
Fator de ruído	2.0 dB
Erro diferencial de propagação máximo	5 ns
Impedância nominal	50Ω
Altitude máxima	9000 m
Características físicas e elétricas	
Dimensões	185 mm diâmetro x 69 mm
Peso	500 g
Potência	
Tensão de entrada	+4.5 a 18.0 VDC
Intensidade de corrente	35 mA
Conexão	fêmea TNC
Ambiente	
Humidade	95 % não condensada
Resistência à água	IEC 60529 IPX7

3.1.1 Instalação do sensor e da antena

A instalação do sensor e da antena é a primeira tarefa prática e estabelece os alicerces do trabalho. Tem como objetivo a obtenção de dados com elevado rigor para utilização ao longo do trabalho e fornecer estes dados a eventuais processos.

Sendo esta uma tarefa de instalação de um equipamento, é necessário criar condições para o acolher abordo do veículo, mais concretamente, é necessário construir um suporte e encontrar um local apropriado. Devido ao reduzido espaço da bagageira do ATLAS-CAR 2, torna-se necessário encontrar um local em que o sensor esteja devidamente

fixado, não acarretasse transtornos na utilização da bagageira e permitisse a passagem do cabo da antena para o tejadilho do veículo.

O suporte foi construído em chapa de alumínio de 2 mm de espessura e o seu desenho técnico pode ser encontrado no anexo A.

De acordo com o manual do sensor (NovAtel Inc., 2013), a sua instalação é uma etapa crítica ao seu bom funcionamento e carece de alguns cuidados. O processo de instalação realizado tem por base os seguintes requisitos:

- A distância e orientação entre o sensor e a antena devem ser constantes.
- A medição da distância entre o sensor e antena, segundo os três eixos (figura 3.2), deve ser efetuada com máximo de rigor possível.

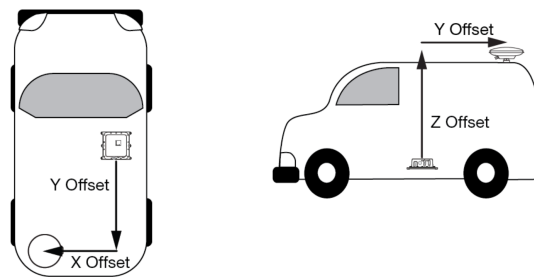


Figura 3.2: Exemplo de instalação e representação da distância entre o sensor e a antena segundo os três eixos (NovAtel Inc., 2013).

- As orientações do sensor segundo os três eixos ortogonais, relativamente à orientação de defeito (figura 3.3), têm de ser conhecidas. De modo a facilitar as medições da orientação do sensor segundo os três eixos, este será colocado com o seu eixo X alinhado com vetor da gravidade, o eixo Z terá a mesma direção do eixo do veículo e o eixo Y aponta para a retaguarda do veículo, figura 3.4e. A falta de rigor desta medição tem como consequência o aumento do tempo de alinhamento na inicialização do sistema.

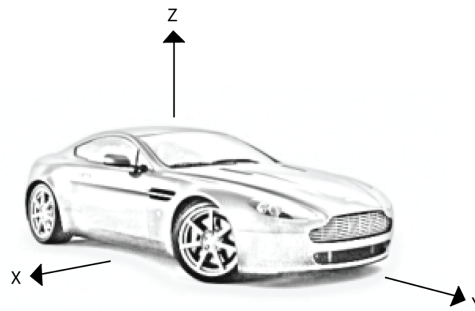
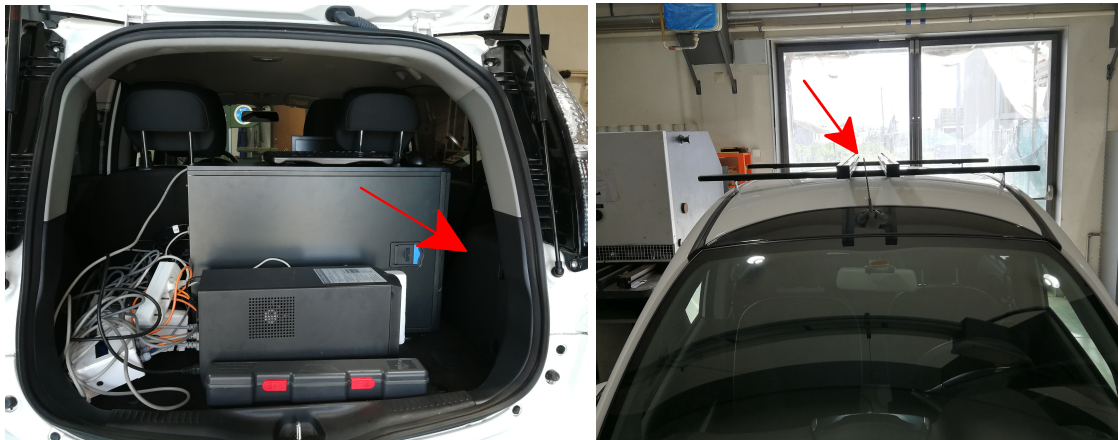


Figura 3.3: Orientação de defeito do sensor (NovAtel Inc., 2013).

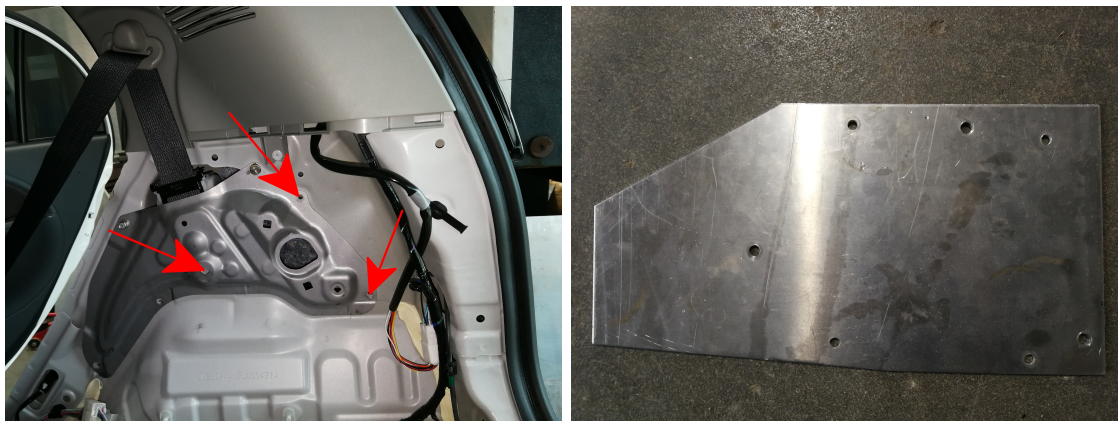
- A unidade inercial deve traduzir as orientações da carroçaria do veículo (figura 3.4c), ou seja, o suporte do sensor deve estar sujeito a uma ligação rígida.

A figura 3.4 mostra as etapas principais do processo de instalação.



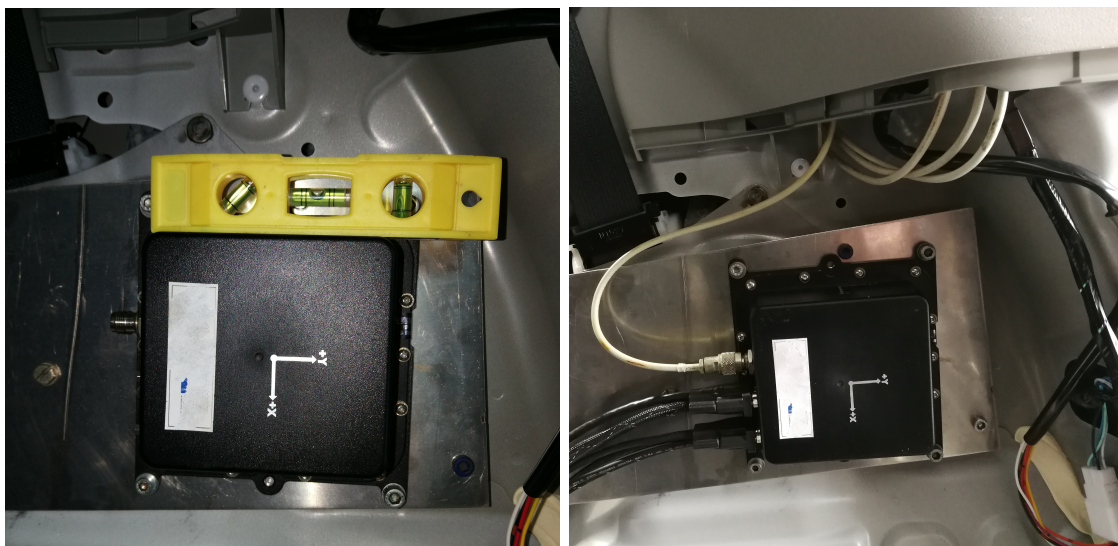
(a) Definição do local de instalação do sensor.

(b) Definição do local de instalação da antena.



(c) Local de fixação do suporte.

(d) Suporte.



(e) Orientação do sensor.

(f) Passagem do cabo co-axial entre o sensor e a antena.

Figura 3.4: Principais etapas da instalação do sistema constituído pelo sensor GNSS+INS e antena.

Na figura 3.5, pode ser visualizado o resultado final da instalação tanto do sensor como da antena.



(a) Finalização da montagem.

(b) Instalação final da antena.

Figura 3.5: Resultado final da instalação do sistema constituído pelo sensor e antena.

3.1.2 Configuração do sensor

A configuração do sensor visa fornecer-lhe informação sobre a disposição do sistema constituído pelo sensor e pela antena, nomeadamente, indicar a distância entre o sensor e antena e a orientação do sensor face à orientação de defeito, fornecer informação sobre a configuração da ligação, por exemplo qual o *baudrate* da comunicação e indicação da forma inicialização da comunicação.

O sensor permite que a comunicação com o utilizador seja realizada segundo vários protocolos, como por exemplo RTCA, NOVATELX, NMEA. Neste trabalho serão utilizados os protocolos NOVATELX e NMEA numa comunicação série *RS 232* com um *baudrate* de 230400 bps. Deste modo, é possível ter acesso a informação *standard*, através do protocolo NMEA, e uma informação mais detalhada através do protocolo NOVATELX, como por exemplo as frequências que estão a ser utilizadas no cálculo da posição e velocidade.

Na figura 3.6 é possível observar a configuração final da instalação do sistema, assim como todas as dimensões envolvidas no processo de configuração.

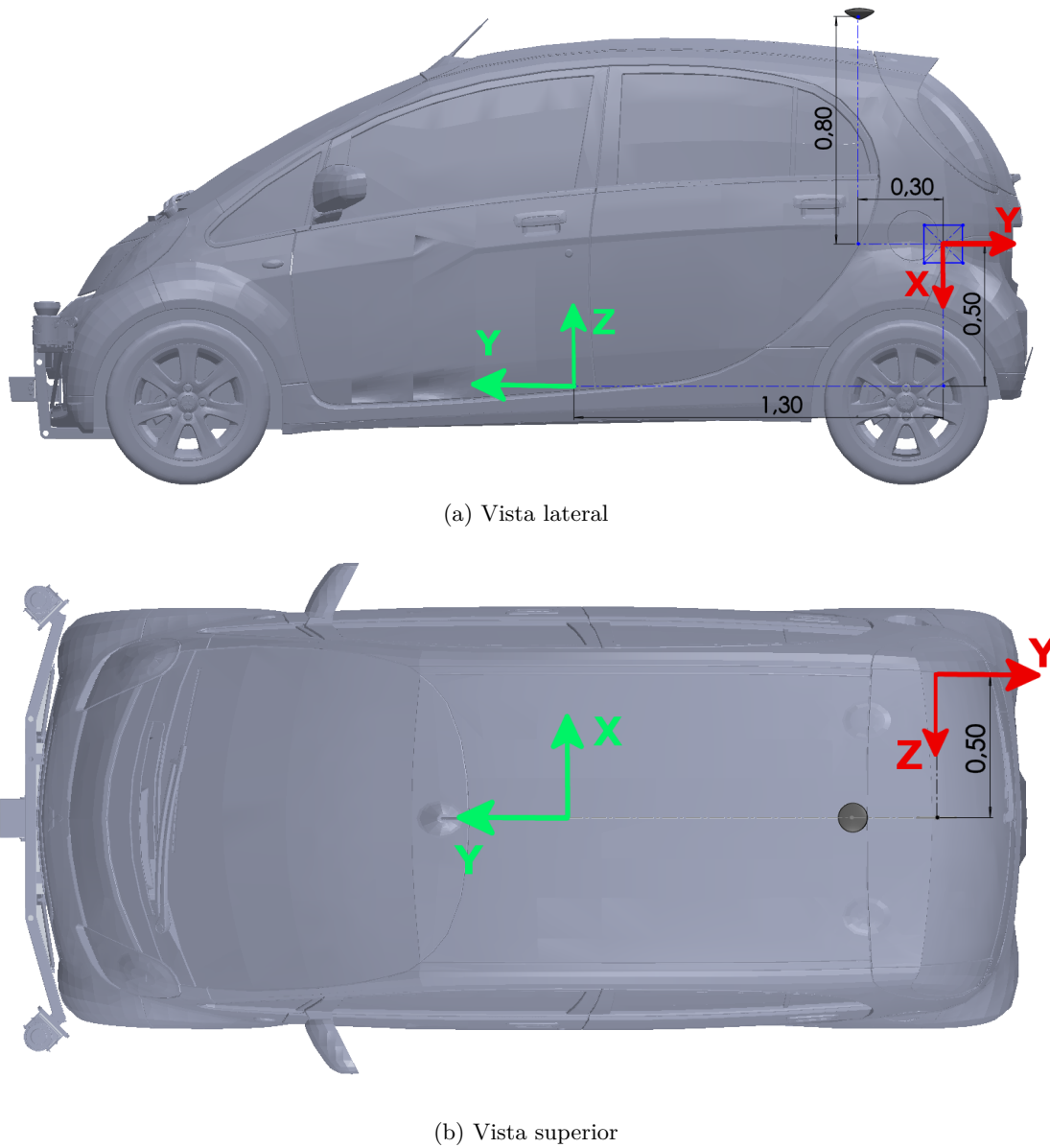


Figura 3.6: Esquema de instalação, a verde o sistema de eixos do veículo e a vermelho o sistema de eixos do sensor.

A tabela 3.3 apresenta todos os comandos necessários para a configuração do sensor. Este processo pode ser realizado graficamente através da aplicação *NovAtel Connect* ou diretamente a partir de um terminal serie, como por exemplo *CuteCom*.

A configuração da orientação é uma etapa bastante complexa; inicialmente é necessário indicar o eixo alinhado com a gravidade, neste caso o eixo X , levando a que $X_i = -Z_{veículo}$, $Y_i = Y_{veículo}$ e $Z_i = X_{veículo}$. De seguida, é necessário proceder a rotação de $\pm 180^\circ$ segundo X_i , contudo esta transformação deve ser introduzida segundo o referencial do veículo, pelo que será introduzida uma rotação de 180° segundo o eixo $Z_{veículo}$ de modo a obter a configuração representada na figura 3.6. O sistema de eixos do veículo foi definido como o centro geométrico do retângulo composto pelas quatro

rodas e à sua cota Z foram somados 10 cm relativamente à altura dos eixos, de modo a compensar a influência da suspensão do veículo.

Tabela 3.3: Comandos de configuração.

Comando	
Protocolo de comunicação	
Configuração das portas	SERIALCONFIG COM1 230400 N 8 1 N ON SERIALCONFIG COM2 230400 N 8 1 N ON
Características físicas da instalação	
Eixo orientado com a gravidade - eixo X	SETIMUORIENTATION 2
Orientação relativa à orientação de defeito (figura 3.3)	VEHICLEBODYROTATION 0.0 0.0 180.0 0.0 0.0 0.1 APPLYVEHICLEBODYROTATION enable
Distância entre o sensor e antena (figura 3.2)	SETIMUTOANTOFFSET -0.80 -0.30 0.50 0.01 0.01 0.01
Origem do veículo	
Distância entre o sensor e o centro do veículo	SETINSOFFSET 0.5 -1.3 0.5
Tipo de alinhamento	ALIGNMENTMODE UNAIDED
Guardar configuração	SAVECONFIG

3.2 Google Maps API Directions

A navegação global, assunto central desta dissertação, necessita de uma aplicação capaz de gerar uma rota entre a posição do veículo e um local de destino. A escolha da aplicação foi realizada tendo como base dois critérios, a fiabilidade da rota, ou seja, os pontos de rota (*waypoints*) devem-se encontrar no interior da faixa de rodagem e principalmente o serviço deve ser *on-line* e sofrer uma grande cadência de atualização, para deste modo, evitar a necessidade de instalar atualizações frequentemente.

Existem vários serviços disponíveis *on-line* capazes de responder às necessidades deste trabalho, como por exemplo, *Google Maps API Directions*, *Open Route Service*, *Mapbox API Direction* e *ViaMichelan*. A escolha recaiu sobre o serviço *on-line* da *Google* pois regista 25 milhões de atualizações diariamente, apresenta cobertura em 99% do território mundial (Google, 2018) e os *waypoints* fornecidos representam com grande fiabilidade a geometria do percurso (figura 3.7).

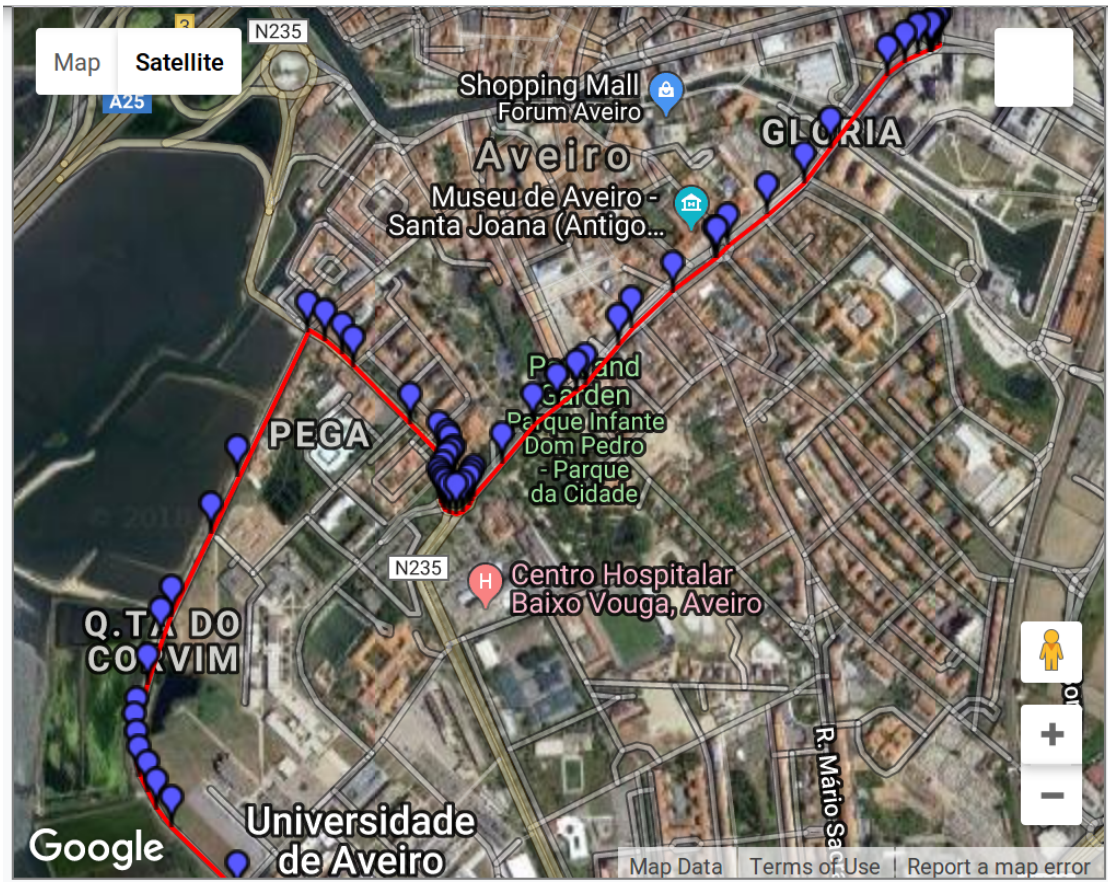
Este serviço permite escolher o tipo de deslocação, automóvel, bicicleta, a pé ou transporte públicos, neste caso será utilizado o tipo automóvel. A determinação da rota é efetuada segundo vários parâmetros como o tráfego atual, distância a percorrer, a utilização ou não de vias com portagens ou *ferry*. Para se ter acesso ao serviço é necessário a criação de uma *API key* com uma conta de pagamento associada.

A estrutura da mensagem de pedido de rota base pode ser observada de seguida;

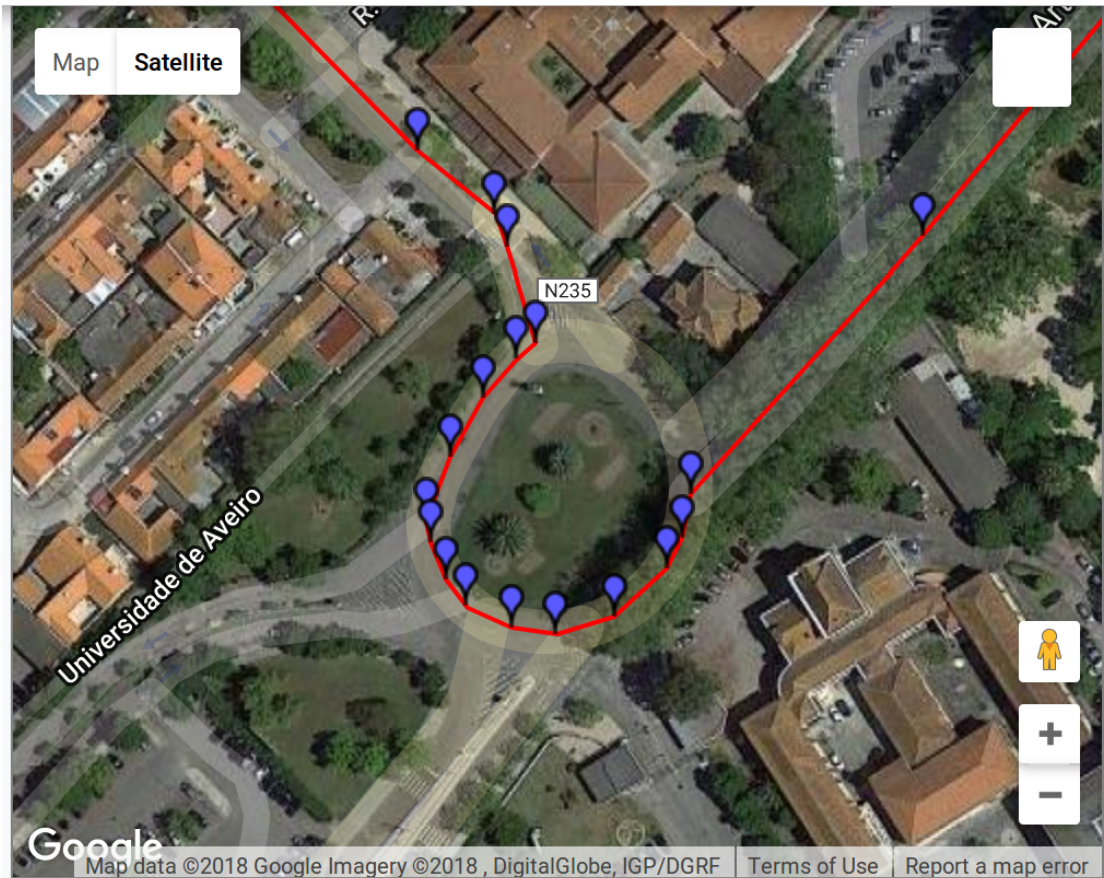
```
https://maps.googleapis.com/maps/api/directions/json?origin=latitude,  
longitude&destination=latitude,longitude&key=****
```

Os pontos assinalados na figura 3.7 foram obtidos a partir da descodificação do parâmetro *overview polyline* de uma rota ilustrativa. Na sua representação, pode-se observar que todos os pontos estão dentro da faixa de rodagem e que existe um grande número de pontos nas zonas de curvas de forma a suavizar as variações de orientação.

No anexo B.1 é apresentado um exemplo da estrutura da mensagem de pedido de rota, e resposta entre dois locais. Sendo o automóvel o meio de transporte, não são pedidas alternativas, nem impostas quaisquer medidas limitativas, como por exemplo evitar vias com portagens. Por defeito, o serviço fornece a rota com a estimativa de tempo de execução mais baixa tendo em atenção variáveis como a velocidade máxima da via ou o volume de tráfego no momento do pedido.



(a) Distribuição de *waypoints* ao longo de um percurso na cidade de Aveiro.



(b) Posicionamento dos *waypoints* no interior da rotunda no percurso anterior.

Figura 3.7: Serviço *on-line* Google Maps API Directions.

3.3 Monitorização

A monitorização é a terceira grande etapa da presente dissertação e tem como objetivo a atualização, em tempo real de uma página *web* com informação relativa ao estado do veículo, nomeadamente o seu posicionamento global e o destino da sua missão. Esta tarefa requer a criação de uma infraestrutura que transmita a informação do *software* presente no veículo até ao servidor do LAR, de modo a ser disponibilizada *on-line*.

Como referido na secção 1.4.1, esta tarefa foi realizada com sucesso no passado com a criação de uma infraestrutura centrada numa base de dados *Postgresql*, acedida remotamente a partir do veículo para fornecimento de informação. Por outro lado, servia como fonte de informação para o processo de disponibilização *on-line* do estado do veículo (figura 3.8).

- Home
- News
- Events
- LAR Meetings
- Atlascar Live!
- Current Student Projects Blogs
- Past Student Projects Blogs
- Masters Thesis Proposals
- Documentation ▶
- Volunteers
- Contacts

ATLASCAR LIVE

ATLASCAR - click to hide/show

Time:	Speed:	RPM:	Steering(deg)	Emergency

ATLASCAR Location - click to hide/show

Latitude	Longitude	Altitude_GPS	Track_GPS

ATLASCAR full set of variables (click to show)

Throttle	Brake	Clutch
Throttle press	Brake press	Clutch press
Auto throttle	Auto brake	Auto clutch
Auto steering	Auto ignition	Gear
Handbrake	Lights minimum	Lights medium

Figura 3.8: Estado inicial da página web.

Contudo, esse trabalho não se encontra funcional desde a evolução da plataforma ROS de *rosbuild* para *catkin make*. A evolução da aplicação *Google Maps* nos últimos anos obriga a adaptações de código para o funcionamento do seu *widget*.

Portanto, fica claro que a realização desta tarefa passa pelo desenvolvimento de uma nova base de dados, estruturada de acordo com as variáveis disponíveis do veículo e desenvolvimento de software que ultrapasse os problemas mencionados. As variáveis introduzidas na base de dados serão a localização e velocidade instantânea do veículos, o destino da missão e parâmetros de organização, como a hora e data.

Nas secções 4.5 e 4.6 são apresentados todos os passos no desenvolvimento da estrutura e processo de escrita de informação na da base de dados e adaptações à página web.

Paralelamente ao processo de monitorização surgiu a ideia da criação de outra base de dados *Postgresql*, localizada no servidor do veículo, com o objetivo de funcionar como memória partilhada por eventuais processos que necessitem, por questões de redundância ou robustez, por exemplo o módulo navegador global possa ser inicializado apenas com a indicação "vai para o último destino". Nesta base de dados serão guardadas as informações de início e fim de missão, todos os *waypoints* obtidos pela decodificação da *polyline* fornecida pelo serviço, assim como as etapas do trajeto e as suas respetivas indicações, como por exemplo "vire à direita".

Capítulo 4

Arquitetura de Software

Este capítulo explica o desenvolvimento e funcionamento dos *packages* ROS que constituem a solução final desenvolvida neste trabalho. Inicialmente são explicados os mecanismos de comunicação com o sensor e o funcionamento da aplicação *Mapviz* como interface com o utilizador. De seguida, é detalhado o funcionamento do *package* responsável pela aquisição e planeamento da rota. Por fim, é apresentado o processo de monitorização implementado.

4.1 *Robot Operating System (ROS)*

A plataforma de desenvolvimento ROS, criada em 2007, é constituída por um conjunto de bibliotecas e ferramentas dedicadas principalmente ao desenvolvimento de aplicações para robôs (Fernandez et al., 2015). A plataforma ROS disponibiliza aplicações capazes de simular e monitorizar as aplicações desenvolvidas, como por exemplo, *Gazebo* ou *Rviz*, e no seu longo repositório é possível encontrar suporte para um elevado número de equipamentos (sensores, câmaras, etc.) ou módulos para a execução de determinadas tarefas (por exemplo, comunicar com um sensor).

Esta plataforma caracteriza-se pela sua arquitetura modular, facilitando a execução de vários processos em simultâneo, e no caso de algum desses processos deixar de funcionar, a aplicação continua a correr. O programador pode escolher a linguagem a adotar no desenvolvimento de *software* utilizando a plataforma ROS, contudo, neste momento, apenas é disponibilizado suporte robusto para as linguagens C++ ou Python.

A plataforma ROS pode ser dividida em três níveis, o *file system level*, *computation graph level* e o *community level*.

O *file system level* corresponde à localização e organização de ficheiros do ROS em disco. Esta organização é realizada através de *meta packages* (conjuntos de *packages*), *packages* (unidade básica na plataforma de desenvolvimento ROS, é constituída por nodos que interagem através de tópicos ou serviços), tipos de serviço ou mensagens (estruturas de mensagens ou serviços).

O *community level* é composto por todas as ferramentas de suporte disponibilizados pelo ROS, por exemplo várias distribuições, repositórios, *wiki*, *ROS Answers* que representam um auxílio muito grande para o utilizador da plataforma.

O *computation graph level* é o nível onde é realizado o procesamento dos programas. Este nível é constituído pelas seguintes partes:

- **Master:** O ROS *Master* efetua o registo e gestão dos nodos. O *Master* deve ser lançado antes dos outros nodos, de forma a que todos os nodos se conheçam e possam trocas mensagens entre si.
- **Nodos:** Normalmente, os nodos são processos ou executáveis responsáveis pela realização de alguma tarefa. Um sistema deve ser composto por vários nodos encarregues de executar tarefas simples (Fernandez et al., 2015). Por exemplo, o processo de adquirir dados de um sensor é a tarefa de um nodo, enquanto que o eventual tratamento desses dados deve ser realizado através de um ou mais nodos, permitindo desta forma, uma grande flexibilidade no desenvolvimento de grandes projetos.
- **Tópicos:** O tópico é um meio de comunicação uni-direcional para a troca de mensagens entre nodos. Para se realizar a troca de mensagens entre nodos é necessário que um nodo publique um tópico e que outro subscreva esse tópico. A arquitetura ROS permite a existência de vários nodos a publicar num tópico, assim como, vários nodos a subscreverem o mesmo tópico (Topics, 2018). É de salientar a importância da designação atribuída ao tópico, a utilização de nomes iguais é geradora de grandes confusões na comunicação entre nodos que usem tópicos do mesmo tipo.
- **Mensagem:** As mensagens são o principal método de comunicação utilizado na troca de informação entre nodos. Existem vários tipos de mensagens padrão disponíveis para utilização na biblioteca ROS, contudo, existe a possibilidade do utilizador proceder à criação de uma mensagem personalizada (Fernandez et al., 2015).
- **Serviços:** Os serviços permitem o estabelecimento de uma comunicação bi-direcional entre dois nodos. Neste método de comunicação existe um nodo que efetua um pedido (cliente) e um nodo servidor que responde. Neste caso existe a necessidade de estabelecer dois tipos de mensagens, uma mensagem de pedido e outra de resposta (Service, 2018).
- **Bags:** São ficheiros que armazenam todas as mensagens de tópicos e serviços, permitindo a sua reprodução posteriormente. Esta ferramenta permite ao utilizador correr os seus programas sem a utilização de *hardware*, evitando desta forma o desgaste dos equipamentos e agilizando o processo de reprodução de testes com os mesmos dados.

Nas próximas secções será apresentado o *software* desenvolvido utilizando a plataforma ROS.

4.2 Comunicação com o sensor

Uma das grandes vantagens da plataforma ROS é a existência de um grande repositório, onde é possível encontrar *drivers* para muitos sensores. No caso do sensor utilizado existem duas possibilidades, a utilização dos *drivers* disponibilizados pelo *GPS Vehicle Dynamics Laboratory* da Universidade de Auburn, que apesar de não terem sido desenvolvidos para o modelo utilizado, funcionam com este modelo (utilizados por Ferreira, 2015), ou a utilização dos *drivers* desenvolvidos pelo *Southwest Research Institute*,

que apresentam grande capacidade, como por exemplo, a possibilidade da comunicação ser realizada no formato ASCII ou binário. A escolha recaiu sobre a segunda opção, pois trata-se de um trabalho mais recente e principalmente porque consegue estabelecer comunicação com base no protocolo NMEA e NOVATELX.

Como mencionado na seção 3.1.2, a comunicação é realizada segundo uma comunicação série *RS-232* com um *baudrate* de 230400 bps. Este valor foi estabelecido com base no conversor *RS-232* - USB utilizado, que apresenta um *baudrate* máximo de 250000 bps. Contudo, devido à grande quantidade informação recebida, informação relativa ao posicionamento a 20 Hz e ao sistema inercial a 100 Hz, este valor de *baudrate* é insuficiente.

Para ultrapassar este problema foram realizadas algumas alterações aos *drivers* utilizados, de modo a possibilitar a comunicação a partir de duas portas séries. Estas alterações têm como objetivo eliminar a publicação de mensagens de defeito, ou seja, os *drivers* utilizados efetuam sempre o envio de mensagens *standard* (protocolo NMEA), contrariamente às mensagens do protocolo NOVATELX que têm associadas a si um parâmetro configurável para o seu envio ou não, pelo que foi necessário criar um parâmetro similar para a mensagens do protocolo NMEA. O excerto de código seguinte, mostra a criação de um parâmetro que controla a publicação, ou não, dos tópicos anteriormente publicados por defeito.

```
namespace novatel_gps_driver{
class NovatelGpsNodelet : public nodelet::Nodelet{
public:
    void onInit()
    { (...)
    swri::param(priv, "publish_default_messages",
        publish_default_messages_, publish_default_messages_);
    (...)
    if (publish_default_messages_)
    {
        gps_pub_ = swri::advertise<gps_common::GPSFix>(node, "gps",
            100);
        fix_pub_ = swri::advertise<sensor_msgs::NavSatFix>(node, "fix
            ", 100);
    }
    (...)
    }

private:
    bool publish_default_messages_;
}}
private:
bool publish_default_messages_;
```

Listagem 4.1: Excerto de código desenvolvido para criar um parâmetro (*publish_default_messages_*) associado à publicação dos tópicos *gps* e *fix*.

Contudo, essa não foi a única alteração aos *drivers* utilizados, foi implementada de raiz uma nova mensagem (*bestgnsspos*). Esta mensagem permite obter a localização global calculada apenas com base no sistema de navegação global por satélite,

isto permite avaliar o desempenho da integração GNSS+INS. No anexo C.1 podem ser observadas as principais etapas no processo de implementação da mensagem *bestgnsspos*.

Assim, o *package novatel_gps_driver* está preparado para lançar dois nodos (*novatel_position* e *novatel_imu*), em que cada um acede a uma porta do sensor.

O nodo *novatel_position* é responsável pela publicação de tópicos relativos ao posicionamento global (figura 4.1). É de salientar que os todos os tópicos relativos ao posicionamento global, com a exceção do tópico *bestgnsspos*, já integram a informação inercial, como explicado anteriormente; essa integração é realizada internamente.

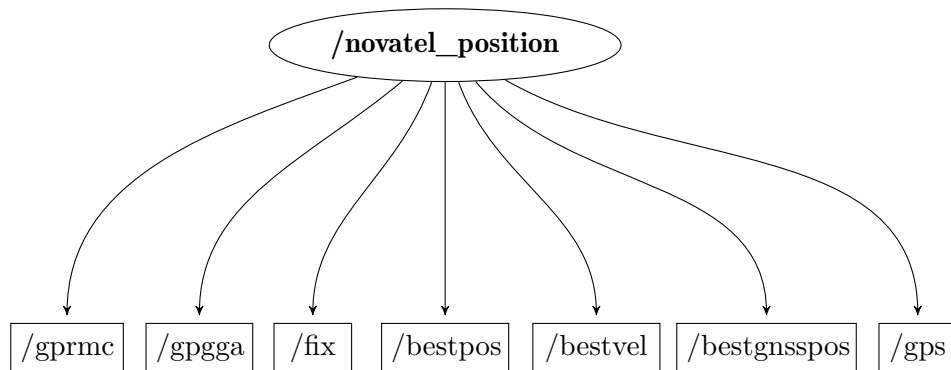


Figura 4.1: Gráfico de distribuição de tópicos para o nodo *novatel_position*.

Por outro lado, o nodo *novatel_imu* é responsável pela publicação de tópicos com informação relativa ao sistema inercial, variáveis como velocidade, *roll*, *pitch*, *azimuth* podem ser encontradas nos tópicos publicados por este nodo (figura 4.2).

A nomenclatura dos tópicos está realizada segundo os protocolos NMEA e NOVATELX. Os tópicos *gprmc*, *gpqga*, *gps* e *fix* apresentam mensagens obtidas através da comunicação segundo o protocolo NMEA, por sua vez, os tópicos *bestpos*, *bestvel* e *bestgnsspos* seguem o protocolo NOVATELX e as variáveis presentes em cada mensagem podem ser encontradas nos manuais NovAtel Inc., 2017 e NovAtel Inc., 2016.

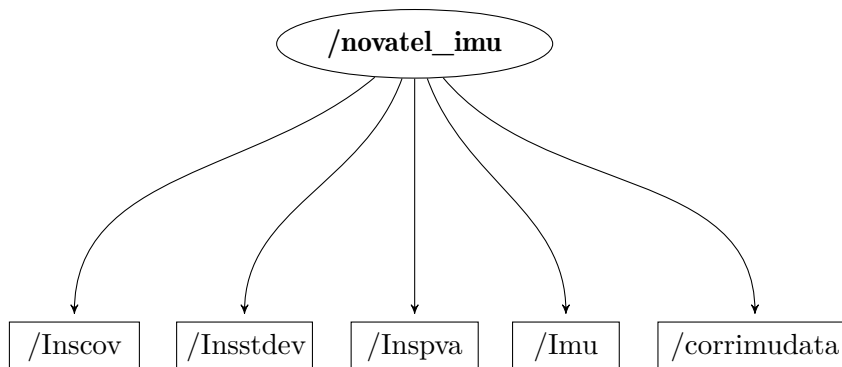


Figura 4.2: Gráfico de distribuição de tópicos para o nodo *novatel_imu*.

No anexo C.2 é possível encontrar o ficheiro *drivers_novatel.launch*, responsável pelo lançamento dos dois nodos.

4.3 Planeamento de rota

O planeamento de rota é um dos componentes principais deste trabalho e tem como objetivo obter uma rota recorrendo ao serviço *Google Maps API Directions* e efetuar o planeamento da rota durante a sua execução.

Para a realização desta tarefa foi desenvolvido um *package* ROS, denominado `global_planning`, com uma estrutura servidor cliente, sendo o servidor responsável pela aquisição de rota através do serviço *Google*, descodificação da mensagem recebida na linguagem JSON. A figura 4.3 descreve o funcionamento do serviço criado.

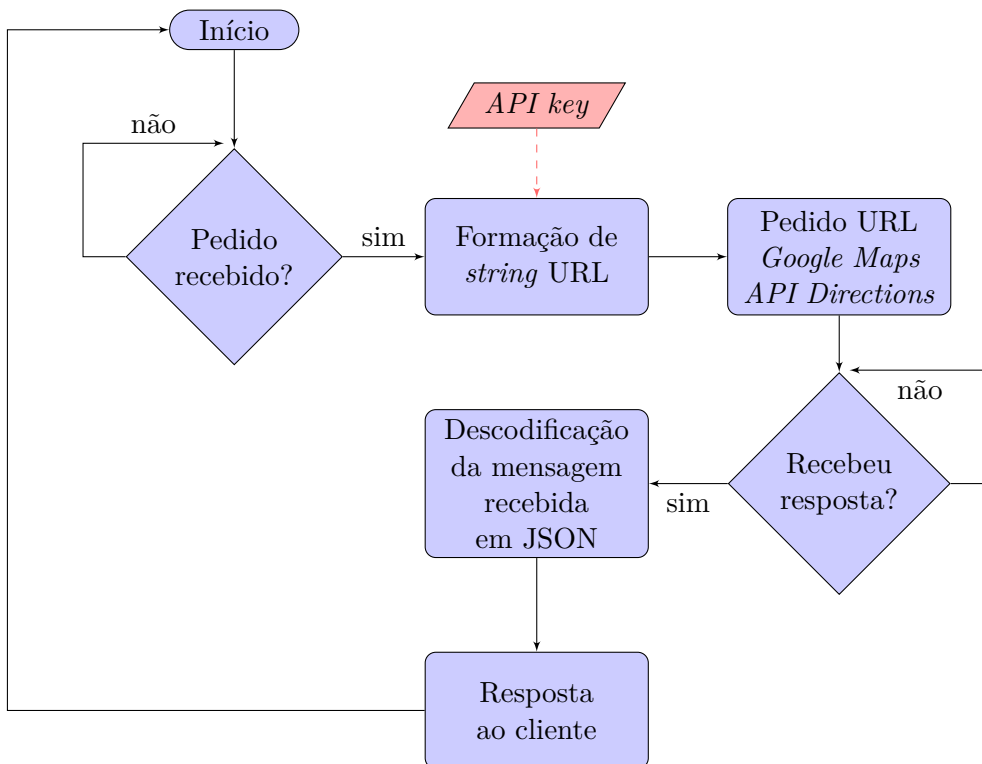


Figura 4.3: Fluxograma de funcionamento do serviço ROS que procede ao pedido de rota junto da aplicação *Google Maps API Directions*.

A mensagem associada ao serviço referente ao pedido é constituída por quatro variáveis, latitude e longitude da posição atual, latitude e longitude da posição de destino introduzida através da aplicação *Mapviz*. A mensagem de resposta é um pouco mais complexa: é constituída por 7 vetores, dois deles destinados às variáveis latitude e longitude de todos os *waypoints* descodificados e os restantes são destinados a definição das etapas da rota, valores de latitude para o início e fim de etapa, assim como para a variável longitude e, por último, a instrução associada a cada etapa.

O nodo cliente é responsável por encontrar continuamente o segmento de caminho que permite ao veículo efetuar corretamente a rota, detetar a saída de rota e proceder ao pedido de um novo serviço, de modo a garantir o sucesso da missão e, não menos importante, detetar a chegada ao destino.

A figura 4.4 mostra a metodologia proposta de planeamento, o retângulo representa o veículo (X), e existe a disposição aleatória de cinco *waypoints* (W_i), que definem segmentos de caminho que o veículo deve ter em conta para chegar ao destino. Os valores de d_i são as distâncias euclidianas entre a posição do veículo e os segmentos de reta. É também, definida uma distância ($d_a = 4m$) como forma flexibilizar o momento de passagem para o próximo segmento.

O processo funciona com base em distâncias euclidianas relativas ao posicionamento global. No exemplo da figura 4.4 o segmento S_2 foi encontrado como sendo o segmento que o veículo deve seguir, com base na distância mínima (d_2) e não se encontrar a uma distância igual ou inferior a 4m do ponto W_3 (ponto final do segmento encontrado).

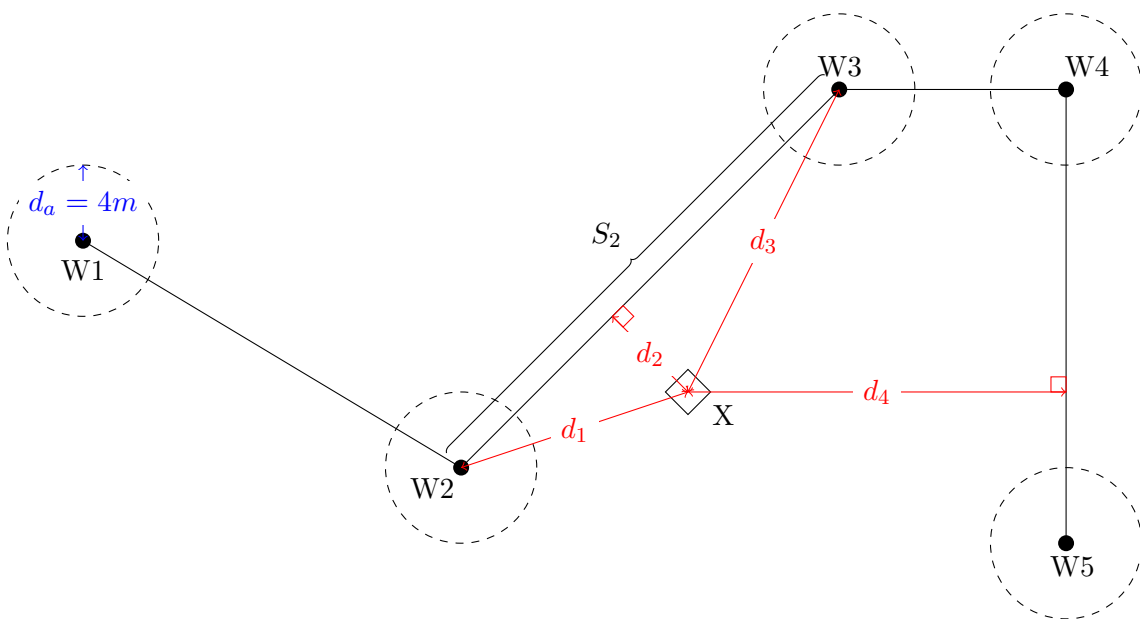


Figura 4.4: Funcionamento da metodologia de planeamento de um segmento de caminho, numa situação genérica.

Na figura 4.5 é apresentada uma situação em que a mínima distância não determina o segmento de referência (S_3), pois o segmento mais próximo está perto do fim e é necessário passar para o seguinte.

Esta metodologia está preparada para que em situações de saída de rota proceda a um novo pedido de rota, para que o veículo tenha uma rota definida até ao destino. Sempre que seja encontrada uma distância mínima d_i superior a 10 m será considerado que o veículo saiu de rota, e é necessário recalculer a rota. Esta é forma de garantir que em situações, por exemplo de obras na via com desvios, que obrigam o planeador local a não cumprir as indicações fornecidas pelo planeador global, e assim o veículo continua a sua missão.

Por fim, é necessário estabelecer uma condição para que seja decretado o fim da missão. Se a distância entre a posição global do veículo e o ponto de destino for igual ou inferior a 5 m será considerado o fim da missão. Este método obriga a que o destino

da missão seja um local acessível ao automóvel, caso contrário não será decretado o final da missão e o planeador local receberá a indicação para continuar para fora de estrada.

A função utilizada no cálculo das distâncias entre a posição absoluta e os segmentos de reta pode ser observada no anexo D.

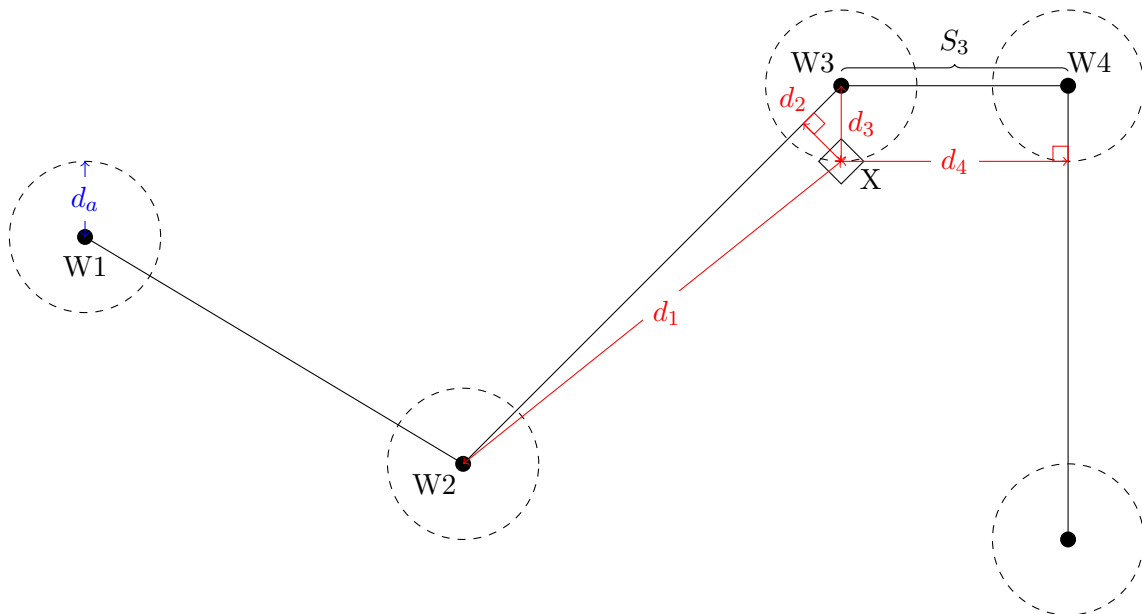


Figura 4.5: Representação gráfica do momento em que ocorre a passagem para o segmento S_3 .

Na figura 4.6 é apresentado o esquema representativo do funcionamento do planeador global.

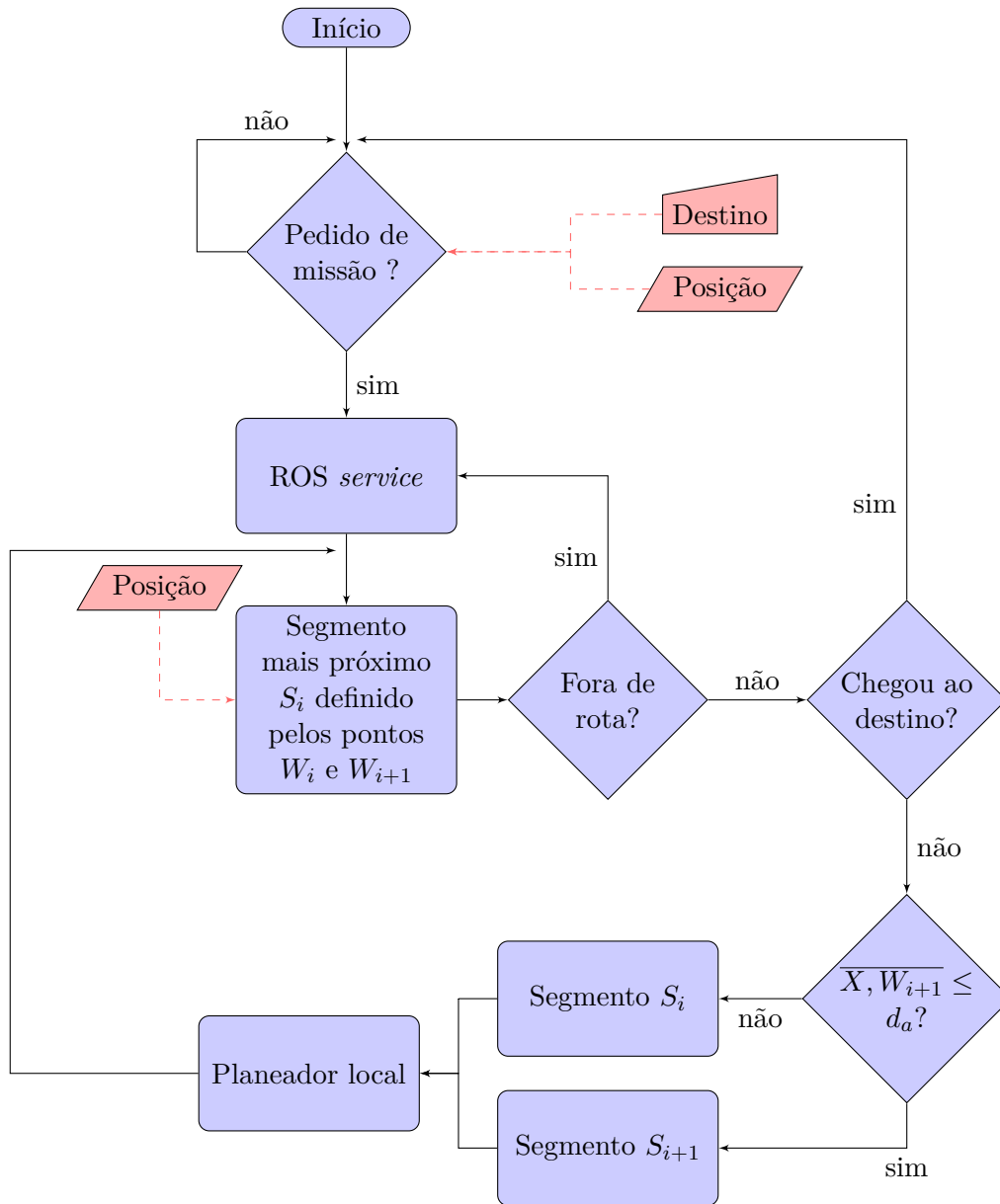


Figura 4.6: Fluxograma de funcionamento do planeador global.

4.4 Aplicação Mapviz

A representação em tempo real do posicionamento global num mapa permite ao utilizador monitorizar o desempenho do sistema. Existem algumas aplicações na plataforma ROS que permitem essa representação, como por exemplo *Gpsd viewer*, *Ground station* ou *Mapviz*. Destas possibilidades, a aplicação *Mapviz* destaca-se claramente das restantes; foi desenvolvida especialmente para aplicações em veículos autónomos e tem desenvolvidos vários *plugins* de forma a criar uma aplicação capaz de agregar informação do posicionamento global, de sensores *lasers*, nuvens de pontos ou câmaras.

A escolha recaiu sobre a aplicação *Mapviz*, pois apresenta características muito abrangentes, como a integração de informação de vários sensores, e pela perspetiva de futuro, pois é de esperar o desenvolvimento de mais *plugins*.

Contudo, a aplicação escolhida apresenta uma grande limitação, apenas contém mapas de baixa resolução e sem vista de satélite. O nível de precisão obtido durante este trabalho levou à necessidade de ter mapas com melhor resolução, com a possibilidade de fazer um grande *zoom* e serem vistas de satélite. Desta forma, reúnem-se condições para a utilização cuidada de todos os *plugins*; por exemplo, é possível monitorizar a mudança de via com maior rigor e as posições de eventuais obstáculos fixos detetados através da análise de uma nuvem de pontos que pode ser comparada com a sua posição no mapa (por exemplo, passeios, limites da via). Para obter os mapas satélite da *Google* segundo o protocolo *Web Map Tile Service (WMTS)*, aplicou-se a metodologia descrita em Snider, 2017, baseada numa ligação através de um *docker*.

De modo a potenciar as características do sistema instalado, foram desenvolvidos dois *plugins*, designados *bestpos* e *bestgnsspos* (figura 4.7). O *plugin bestpos* utiliza a leitura de posição global e valor de orientação (*azimuth*) fornecido pelo sistema inercial, tendo como principal vantagem a utilização dessa orientação. No anexo E.1 pode-se encontrar as principais etapas da implementação do *plugin bestpos*, idêntico ao *plugin bestgnsspos*.

Os *plugins* originais utilizavam apenas o sistema de localização global como fonte do valor do orientação, e em situações de baixa velocidade, essa orientação não está correta. Este desenvolvimento, permite estabelecer um sistema de eixos capaz de ser integrado com todos os sensores instalados no veículo.

O *plugin bestgnsspos* tem como objetivo a representação das leituras de localização sem a interferência do sistema inercial, ou seja, o sensor fornece também o posicionamento sem utilizar a informação inercial e com a sua representação permite observar o desempenho da integração entre o sistema GNSS com o sistema inercial (figura 4.8).

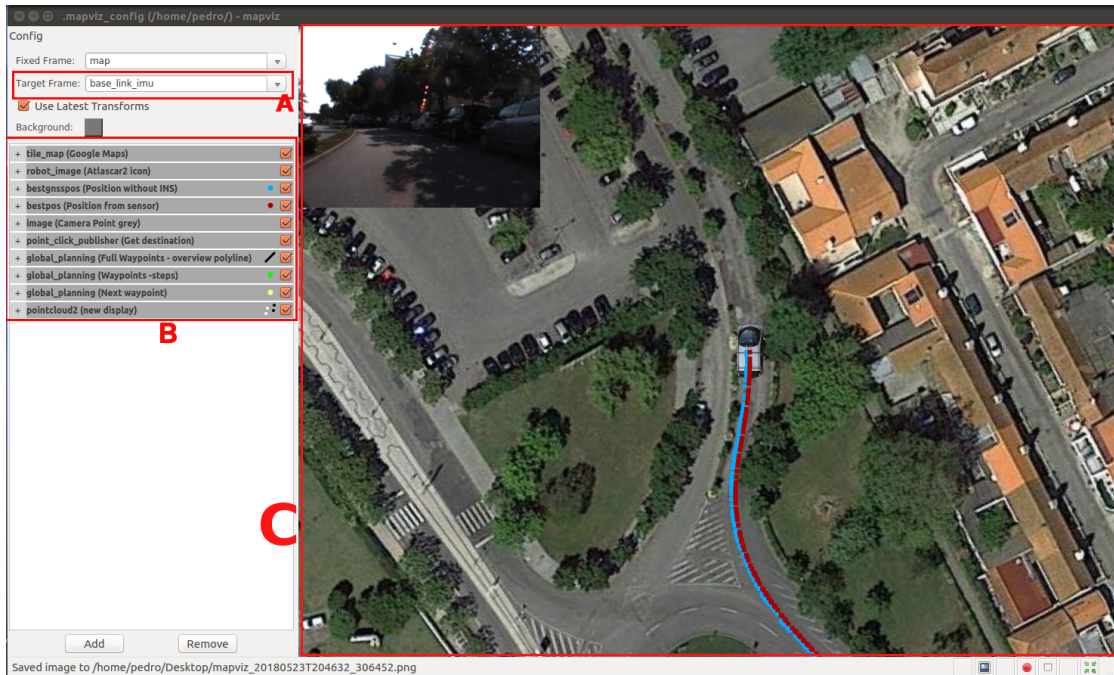


Figura 4.7: Arquitetura da aplicação *Mapviz*. *A* - Escolha do referencial do veículo; *B* - *Plugins* em utilização; *C* - Representação gráfica dos *plugins* em utilização.



Figura 4.8: Representação na aplicação *Mapviz* da posição a vermelho utilizando a integração GNSS+INS e a azul a posição calculada através do sistema GNSS.

A aplicação *Mapviz* é um elemento preponderante na arquitetura da solução apresentada, não só porque permite monitorizar o seu funcionamento, mas também porque permite ao utilizador introduzir o destino da missão. Isto é, o pedido de rota é realizado

através da aplicação recorrendo ao *plugin point_click_publisher* com o nome do tópico, sendo obrigatoriamente "*destination*". É de salientar, que foi necessário a criação de um tópico com a designação "*destination*", que fosse publicado na plataforma ROS sem a utilização de um prefixo variável, que dificulta a sua subscrição.

```
void PointClickPublisherPlugin::topicChanged(const QString& topic)
{(...)
destination_publisher_ = node_teste.advertise<geometry_msgs::
    PointStamped>("destination", 1000);
(...)}
```

Listagem 4.2: Excerto de código onde ocorre a declaração do tópico que contém a posição de destino.

Foi também criado um *plugin* capaz de representar a rota fornecida pelo serviço *Google Maps API Directions* e os dois *waypoints* que definem o segmento de caminho enviado ao módulo de navegação local. É denominado *global_planning* e pode subscrever os tópicos *waypoints_full*, contendo todos os pontos obtido através da descodificação do parâmetro *overview_polyline*, o tópico *waypoints_prev_next*, contendo os dois pontos que definem o segmento de caminho fornecido ao módulo de navegação local. E ainda, o tópico *waypoints_steps*, que contém os pontos de início e de fim de etapa que têm associado uma instrução, por exemplo "vire à direita". É de salientar contudo que estas etapas não são utilizadas no planeamento da rota.



Figura 4.9: Exemplo do funcionamento do *plugin global_planning*, representando a preto os segmentos de retas definidos pelos *waypoints*, a verde as etapas do percurso e a amarelo os pontos que definem o segmento de referência para o veículo.

A figura 4.10 mostra a utilização dos *plugins* relativos à rota e posição do veículo, nuvem de pontos, *icon* do veículo e imagem da câmara frontal do veículo.

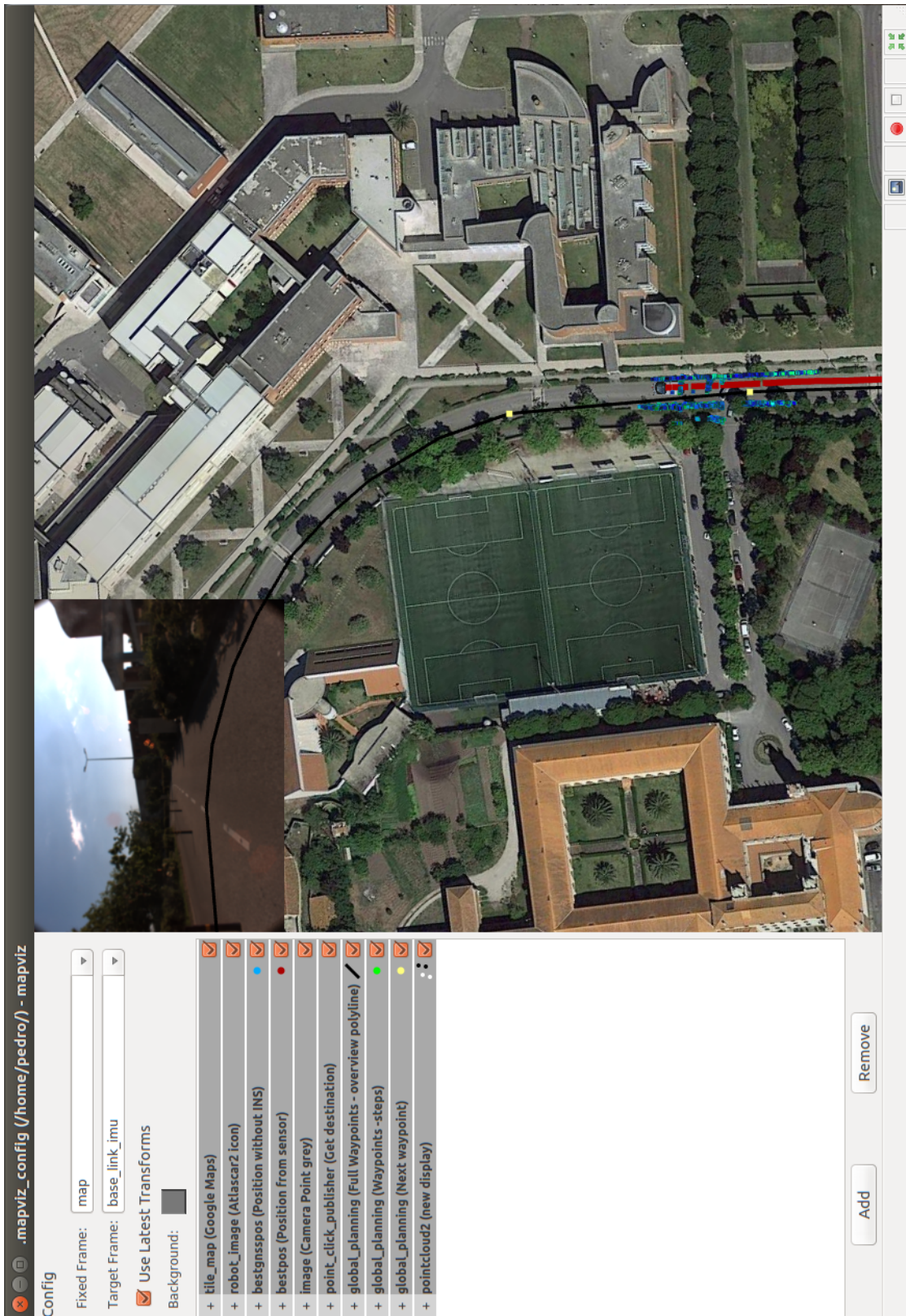


Figura 4.10: Imagem da aplicação *Mapviz* durante o planeamento e execução de uma rota.

4.5 Postgresql

Como referido na secção 3.3, foram desenvolvidas duas bases de dados *Postgresql*, uma localizada a bordo do veículo e outra no servidor do LAR. Estas bases de dados são escritas através do *software* desenvolvido na plataforma ROS. Nas secções 4.5.1 e 4.5.2 será detalhado o desenvolvimento e funcionamento de cada uma.

4.5.1 Base de dados local

A base de dados local tem como objetivo funcionar como memória partilhada por eventuais processos que necessitem. Relativamente à navegação global, permite criar um histórico de destino de modo a que o utilizador possa escolher um destino recente. A base de dados criada em *Postgresql* é designada *atlas_navegacao_global* e as credenciais de acesso são as seguintes:

```
PGconn *conn = PQconnectdb("user=pedro password=*** dbname=atlas_navegacao_global hostaddr=127.0.0.1 port=5432");
```

Para acolher os dados da navegação global foram desenvolvidas três tabelas. Uma tabela destinada aos locais de destino, outra à lista de *waypoints* descodificados, e por último uma tabela destinada às etapas da rota com a instrução associada. As variáveis de cada tabela, assim como o seu tipo e descrição podem ser observados nas tabelas 4.1, 4.2 e 4.3.

Tabela 4.1: Tabela *destination_table* para registo dos locais de destino.

Variáveis	Tipo	Descrição
dest_id	PK <i>text</i>	<i>String</i> com data e hora do pedido
dest_lat	<i>numeric</i>	Valor de latitude do local de destino
dest_lon	<i>numeric</i>	Valor de longitude do local de destino

Tabela 4.2: Tabela *waypoint_table* para registo dos waypoints descodificados.

Variáveis	Tipo	Descrição
waypoint_id	PK <i>serial</i>	Índice da posição
way_lat	<i>numeric</i>	Valor de latitude dos pontos descodificados
way_lon	<i>numeric</i>	Valor de longitude dos pontos descodificados

Tabela 4.3: Tabela *stepswaypoint_table* para o registo dos locais de início e fim de etapa e a sua instrução associada.

Variáveis	Tipo	Descrição
stepwaypoint_id	PK <i>serial</i>	Índice da etapa
start_way_lat	<i>numeric</i>	Valor de latitude dos pontos de início de etapa
start_way_lon	<i>numeric</i>	Valor de longitude dos pontos de início de etapa
end_way_lat	<i>numeric</i>	Valor de latitude dos pontos de fim de etapa
end_way_lon	<i>numeric</i>	Valor de longitude dos pontos de fim de etapa
instruction	<i>text</i>	Instrução associada à etapa

4.5.2 Base de dados remota

A base de dados remota, localizada no servidor do LAR tem como objetivo acolher informação para ser representada na página *web*. Para realizar esta tarefa foi desenvolvido um *package* ROS para proceder à escrita de informação na base de dados. Este *package* é denominado **database** e procede à escrita de informação na base de dados *atlas_monitoring*, que é constituída por uma tabela (tabela 4.4) que agrega variáveis relativas à localização e navegação global. A estrutura da base de dados é projetada para a existência de várias tabelas, de forma a agregar informação por grupos, esta é a forma que deve ser adotada em futuros desenvolvimentos.

As credenciais de acesso à base de dados são as seguintes:

```
PGconn* conn = PQconnectdb("user=atlas password=*** dbname=atlas_monito
ring hostaddr=193.137.172.18 port=5432");
```

Tabela 4.4: Tabela *global_navigation* para o registo da informação relacionada com o posicionamento e navegação global.

Variáveis	Tipo	Descrição
id	PK <i>serial</i>	Índice da leitura na missão
lat_gps	<i>numeric</i>	Valor de latitude da posição atual
lon_gps	<i>numeric</i>	Valor de longitude da posição atual
alt_gps	<i>numeric</i>	Valor de altitude da posição atual
track_gps	<i>numeric</i>	Valor de orientação do veículo
speed	<i>numeric</i>	Valor de velocidade instantânea
lat_dest	<i>numeric</i>	Valor de latitude da posição de destino
lon_dest	<i>numeric</i>	Valor de longitude da posição de destino
date	<i>date</i>	Data da missão
time	<i>time with time zone</i>	Hora da leitura

O *package database* é constituído por um nodo, que utiliza a biblioteca **libpq** escrita em C++ para fazer interagir com a base de dados. A figura 4.11, mostra a arquitetura do funcionamento deste *package*, onde o tópico **/global_destination** contém informação relativa ao destino da missão, os tópicos **/bestpos** e **/inspva** contém informação relativa ao posicionamento global e o tópico **/bestvel** contém a velocidade linear do veículo.

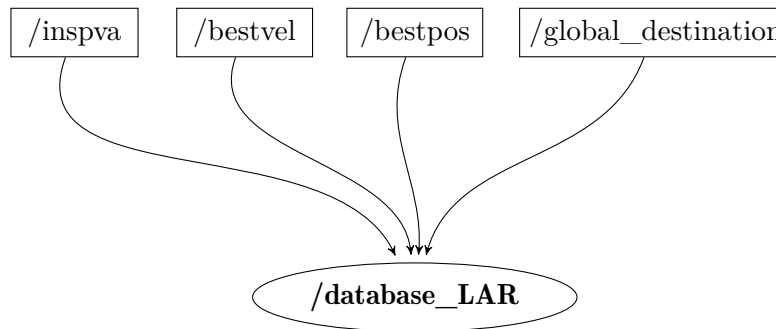


Figura 4.11: Arquitetura de funcionamento do *package database*.

4.6 Página web

A página *web* permite a monitorização remota da localização e do estado do veículo. Desta forma, existe a possibilidade de acompanhar o desenrolar das missões efetuadas pelo veículo ATLASCAR 2.

Como referido na secção 4.5.2, a base de dados remota contém toda a informação disponível para ser mostrada na página *web*. Contudo, existe uma estrutura de ficheiros que possibilitam o transporte de informação da base de dados até à página, nomeadamente um ficheiro "php", um ficheiro "javascript" e um ficheiro "html" (figura 4.12).

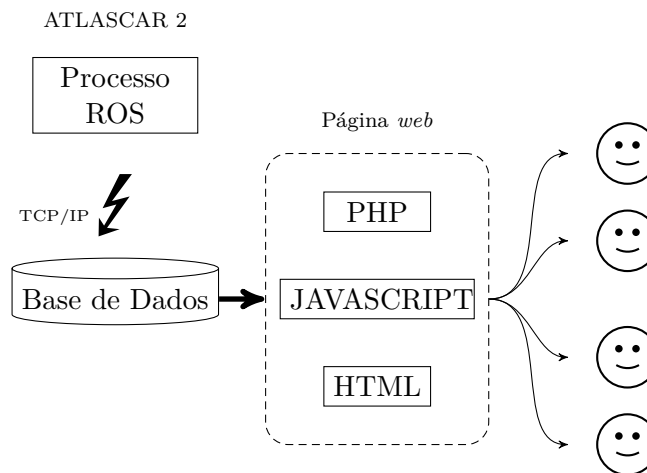


Figura 4.12: Esquema de implementação da monitorização remota do ATLASCAR 2 (adaptado de Viana, 2013).

O ficheiro escrito em linguagem "php" tem como tarefa ler a base de dados e encontrar os últimos valores introduzidos. Neste ficheiro foi necessário configurar as novas credenciais de acesso e ler os valores das variáveis disponíveis na base de dados.

O ficheiro "javascript" tem como responsabilidade gerir o *widget* do *Google Maps*, ou seja utilizar os valores recentes de localização (provenientes do ficheiro "php") e atualizar a localização no *widget*, neste ficheiro foi atualizada a estrutura de comunicação com o serviço *Google*, de forma a contemplar uma *API key*, para que o mapa do *widget* tenha

melhor resolução.

O endereço utilizado na comunicação é o seguinte:

`https://maps.googleapis.com/maps/api/js?key=***`

Por último, o ficheiro "html" trata do *design* da página; foram realizadas alterações de modo a retirar os campos de variáveis que neste momento não estão disponíveis, e criar novos campos para as variáveis que não estavam contempladas na antiga página, como por exemplo o destino da missão.

A figura 4.13 mostra o *design* e funcionamento da nova página *web*.

The screenshot shows a web interface for ATLASCAR 2. On the left is a sidebar with navigation links: LAR Meetings, Atlascar Live!, Current Student Projects Blogs, Past Student Projects Blogs, Masters Thesis Proposals, Documentation, Volunteers, and Contacts. The main content area displays 'ONLINE' in green. Below this is a table for ATLASCAR data:

ATLASCAR - click to hide/show			
Time:	Speed:	Latitude Destino:	Longitude Destino:
23:13:33+01	NA	0.000000	0.000000

Below the first table is another table for ATLASCAR Location:

ATLASCAR Location - click to hide/show			
Latitude	Longitude	Altitude	Track
40.629204	-8.659785	9.029600	355.004416

The bottom section of the page features a Google Maps satellite view of the University of Aveiro campus. A car icon is positioned on the map, and various campus buildings and streets are labeled, including 'R. da Pega', 'Library Biblioteca da Universidade de Aveiro', 'Post Office Estação dos Correios da Universidade de Aveiro', 'Reitoria', 'Departamento de Física da Univer...', 'CICE Insti...', 'Santiago Residence Ria', 'DeCA - Departamento de Comunicação e Arte da...', and 'Rua de S. Tiago'. The map includes standard Google Maps controls like zoom in/out and a person icon.

Figura 4.13: Página *web* da monitorização remota do ATLASCAR 2.

Capítulo 5

Testes e resultados

Neste capítulo são apresentados os resultados obtidos nos vários testes realizados, de modo a validar o sistema global desenvolvido. Inicialmente, são mostrados os resultados relativos ao processo de calibração do sensor em duas situações distintas, de seguida é analisado o desempenho do sensor em situações críticas e por fim é avaliada metodologia de planeamento e execução de rota.

5.1 Calibração do sensor

Como referido na secção 3.1, o sensor utilizado integra um sistema de navegação inercial com o sistema global de navegação por satélite. Esta fusão obriga, no momento da inicialização do sensor, a um processo de calibração, de modo a obter uma convergência no cálculo da posição. O processo de calibração do sensor é um processo extremamente importante, pois permite ao sistema inercial encontrar a sua posição zero relativa.

O processo de alinhamento ou calibração é realizado sempre que o sensor seja ligado. Para que ocorra a convergência da solução é necessário que o veículo se movimente em linha reta por um determinado período de tempo. O período de tempo considerado como período de alinhamento começa com o início do movimento do veículo e termina quando a solução converge e o sensor começa a fornecer as variáveis relativas ao sistema inercial, como por exemplo *roll*, *pitch* e *azimuth*.

Para avaliar o tempo necessário para a convergência da solução foram realizados dois testes. O primeiro foi realizado em campo aberto, onde não existem problemas de reflexão dos sinais dos satélites, e o segundo num ambiente urbano, onde os fenómenos de reflexão originam dificuldades adicionais no processo de convergência e normalmente existe um menor número de satélites disponíveis.

5.1.1 Campo aberto

A realização do alinhamento em condições favoráveis foi realizado no parque de estacionamento da Escola Superior de Saúde da Universidade de Aveiro, local que reúne condições propícias à observação de um elevado número de satélites e onde não ocorre fenómenos de reflexão.

A figura 5.1 mostra o número de satélites observados e utilizados durante o período de alinhamento. Apesar de neste caso o número de satélites observados e utilizados

ser igual, o número de satélites usados pode efetivamente ser inferior ao observado. O alinhamento demorou 9.30 segundos e foram utilizados 10 satélites durante esse período.

Como referido, número de satélites usados e observados pode ser diferente, contudo, neste caso esse valor é constante durante o período de alinhamento, o que comprova as excelentes condições em que o processo de calibração foi realizado.

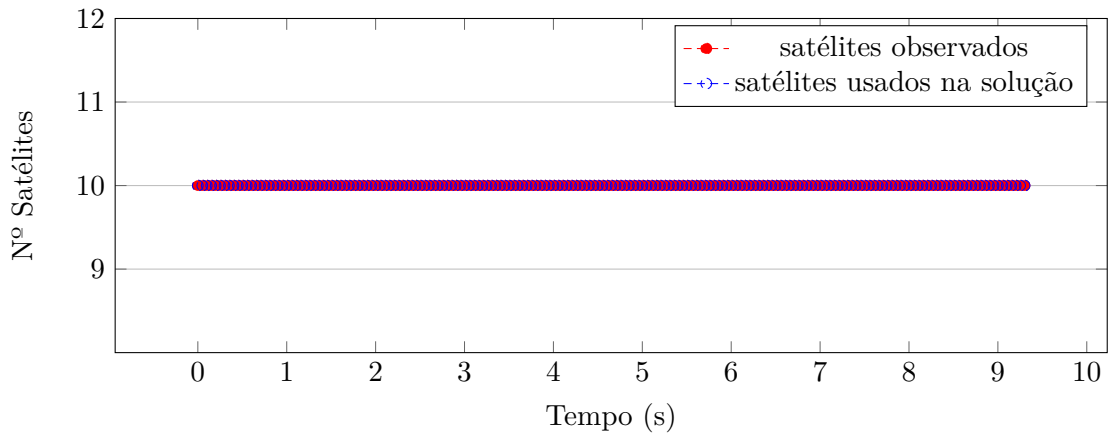


Figura 5.1: Número de satélites ao longo do período de alinhamento em condições favoráveis.

De forma a avaliar a precisão das leituras de posicionamento durante o processo de alinhamento no plano 2D, foi utilizada a equação 5.1 para obter o valor da precisão da posição global, apenas com base no sistema GNSS, dado a indisponibilidade do sistema inercial durante o processo de alinhamento. Esta equação utiliza os valores de desvio padrão da latitude e longitude fornecidos pelo sensor, e o resultados podem ser observados na figura 5.2.

$$\text{erro2D} = \sqrt{\sigma_{\text{latitude}}^2 + \sigma_{\text{longitude}}^2} \quad (5.1)$$

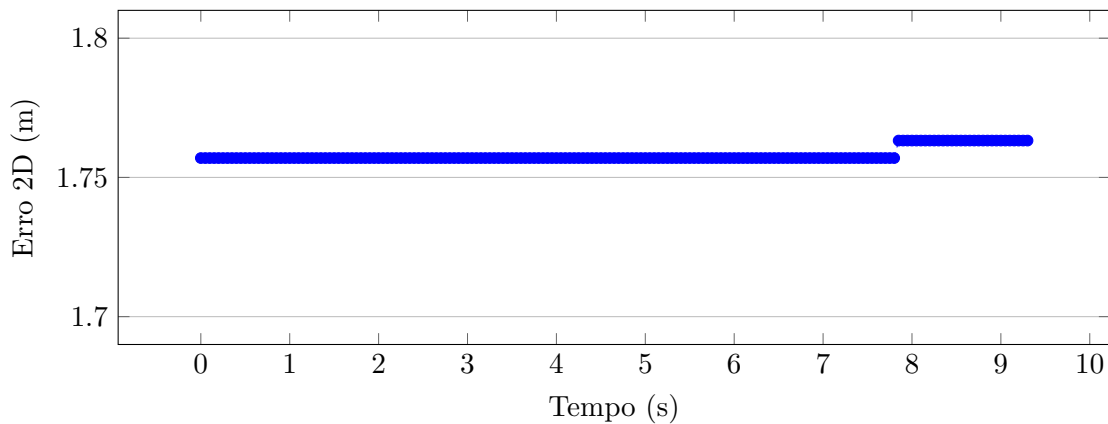


Figura 5.2: Valor de erro das leituras fornecidas pelo sensor durante o processo de calibração em condições favoráveis.

A figura 5.3 mostra a representação das leituras de posição absoluta durante o processo de calibração num mapa em vista de satélite.

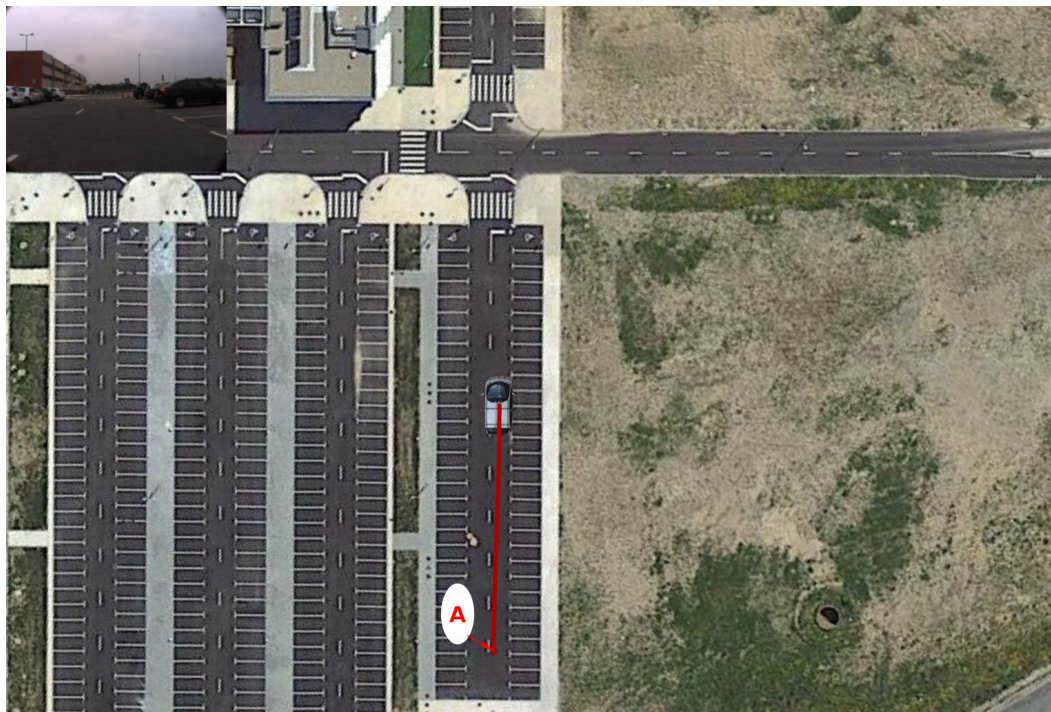


Figura 5.3: Percurso percorrido durante o processo de alinhamento com início no ponto A e fim no local onde o veículo se encontra.

5.1.2 Ambiente urbano

O processo de calibração em ambiente urbano foi realizado na Rua Prof. Manuel Est. Silva, em Aradas, Aveiro (figura 5.4). Esta rua tem apenas uma via e é delimitada por edifícios com 5 andares, onde facilmente ocorrem fenómenos de reflexão que dificultam a convergência do posicionamento.



Figura 5.4: Rua Prof Manuel Est. Silva em Aradas, Aveiro

Este teste foi realizado em 11.8 segundos e na figura 5.5 pode-se observar o percurso efetuado desde o início até à conclusão do processo de alinhamento .

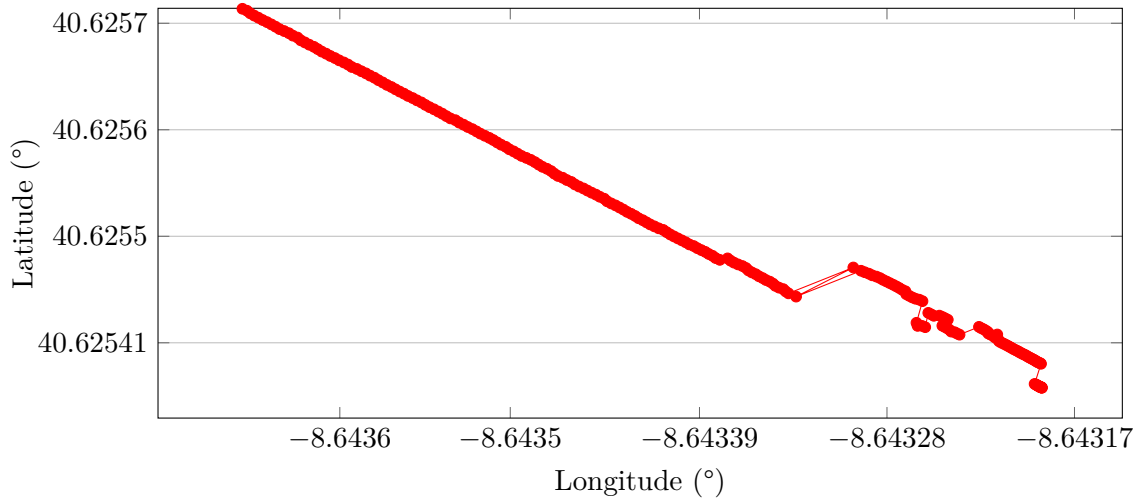


Figura 5.5: Leituras de posicionamento global durante o processo de alinhamento com início no ponto de coordenadas $(-8.643189, 40.625362)$.

Na figura 5.6, pode-se observar a variação do número de satélites ao longo do processo de convergência. É salientar a diferença entre o número de satélites observados e utilizados. Esta diferença comprova o nível de exigência que o local impõe ao sistema.

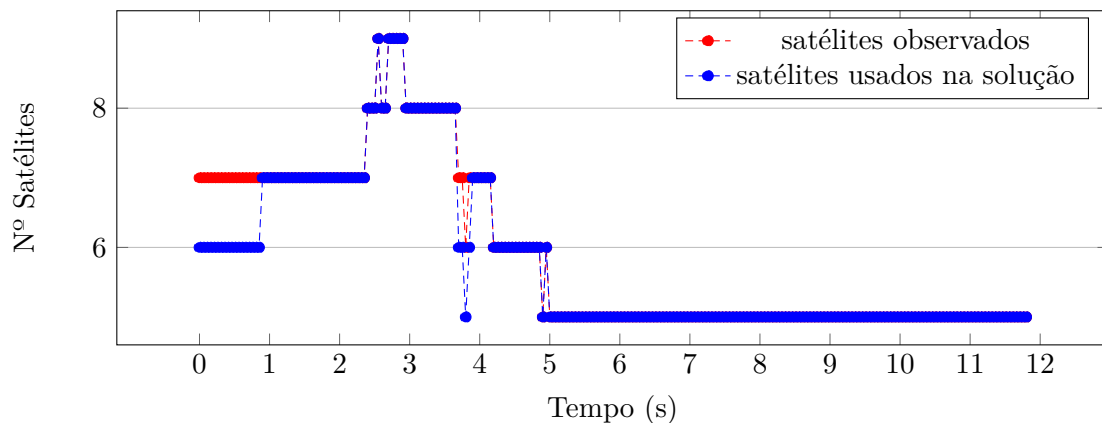


Figura 5.6: Número de satélites ao longo do período de alinhamento em rua delimitada por edifícios.

Na figura 5.7 é apresentado o valor do erro de localização calculado através da equação 5.1. Estes valores encontram-se de acordo como o número de satélites usados, pois, quanto maior o número de satélites menor é o valor de erro. Contudo, a partir dos 5 segundos existe um aumento considerável do valor de erro; este aumento é explicado pela utilização apenas da frequência L1 em todos os satélites, o que impede a aplicação da correção de propagação de sinal na ionosfera com base na comparação das duas frequências (L1 e L2), habitualmente disponíveis. Como referido na secção 2.1, a propagação ao

longo da ionosfera é um aspeto crítico no funcionamento do sistema de navegação por satélite que pode induzir erros de vários metros.

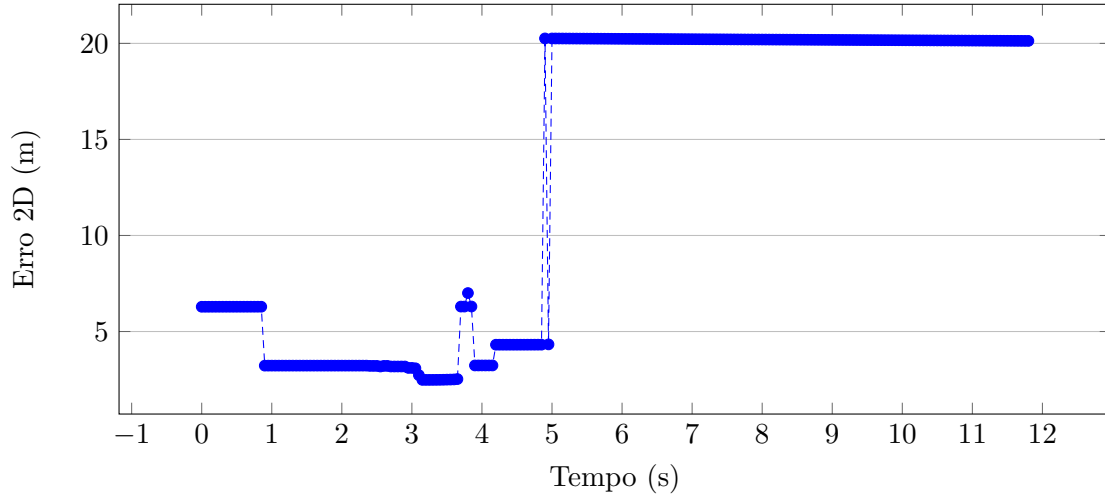


Figura 5.7: Valor de erro durante o processo de alinhamento inicial.

Apesar do valor elevado de imprecisão durante quase todo o processo de alinhamento, isto não se traduz em leituras de posicionamento que impedem o bom funcionamento do veículo. A figura 5.8 mostra a representação num mapa do percurso efetuado, onde se pode observar que todas as leituras se encontram dentro da faixa de rodagem.

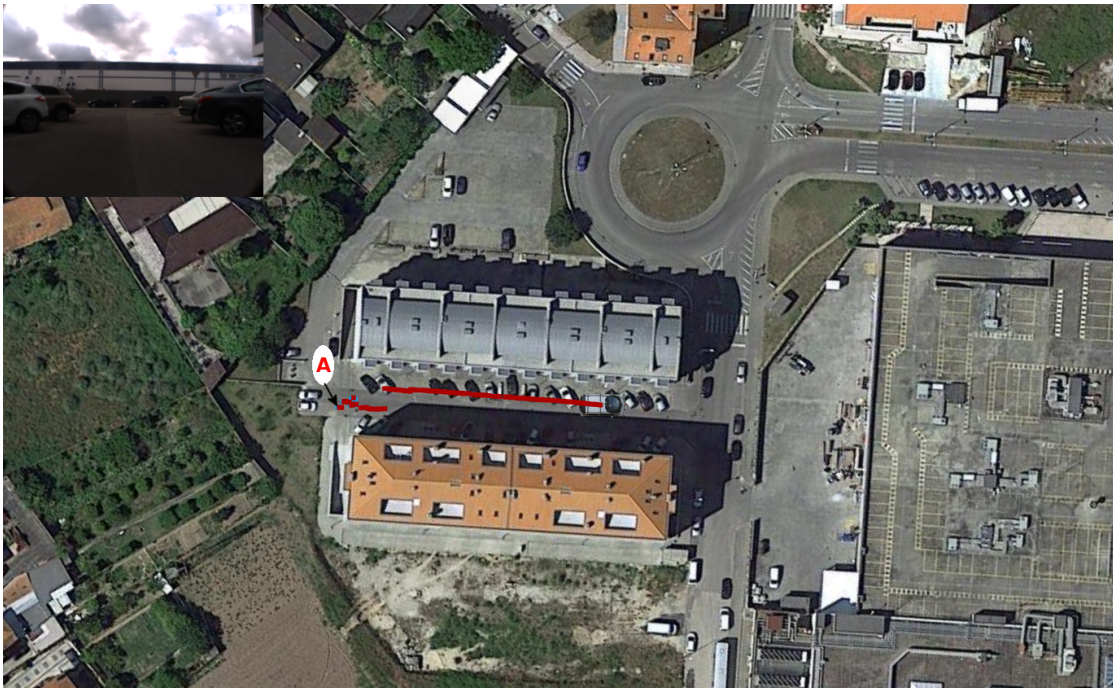


Figura 5.8: Percurso efetuado durante o processo de alinhamento com início em A.

5.2 Situações propícias a falhas do sistema

Com a instalação do sistema de localização global, descrito anteriormente, pretende-se a obtenção de valores com elevado nível de precisão de posicionamento global. Depois de caracterizado o processo de inicialização do sistema, interessa conhecer a sua resposta em situações comuns, quer em ambientes urbanos quer em ambientes rurais.

Para avaliar o desempenho do sistema instalado, foram realizados vários testes de modo a conhecer o nível de fiabilidade do sensor em determinadas situações. Foram escolhidas três situações distintas, duas com características essencialmente de ambientes urbanos, uma rua delimitada por edifícios e túnel. Por fim, são apresentados os resultados relativos à passagem numa rua delimitada por árvores de grande porte, que apesar de se localizar no centro de uma cidade, apresenta grandes semelhanças com ambientes rurais com área florestal.

5.2.1 Rua delimitada por edifícios

Em ambientes urbanos é comum que as ruas estejam rodeadas por edifícios. Como explicado na secção 2.1, os cenários urbanos constituem um grande desafio aos sistemas de posicionamento global. Para avaliar o desempenho do sistema instalado foram realizadas três passagens na Rua Prof. Manuel Est. Silva, duas no sentido noroeste/sudeste (1ª e 3ª passagens) e uma no sentido sudeste/noroeste (2ª passagem).

A figura 5.9 mostra o nível de repetibilidade obtido nas experiências realizadas. Nestas experiências procurou-se conseguir que a rota do veículo fosse a mais idêntica possível, as leituras de posição fornecidas pelo sensor encontram-se de acordo com essa tentativa.

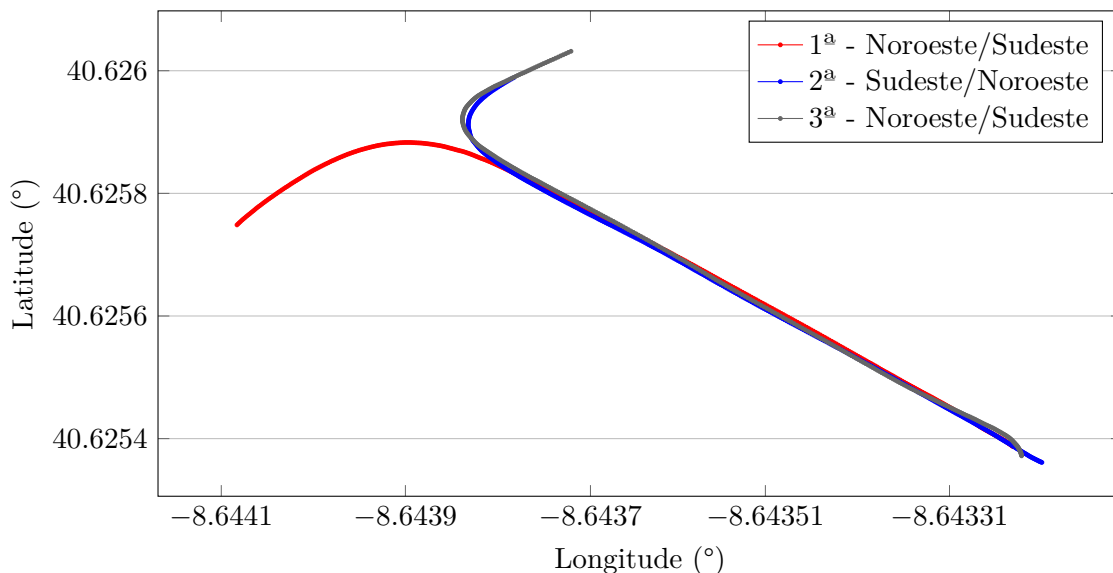


Figura 5.9: Representação da posição global durante as três passagens na Rua Prof. Manuel Est. Silva.

Na figura 5.10 é possível observar o número de satélites utilizados e a velocidade do veículo durante as três passagens. Desta forma, é possível estabelecer uma referência

temporal com a execução do trajeto representado na figura 5.9.

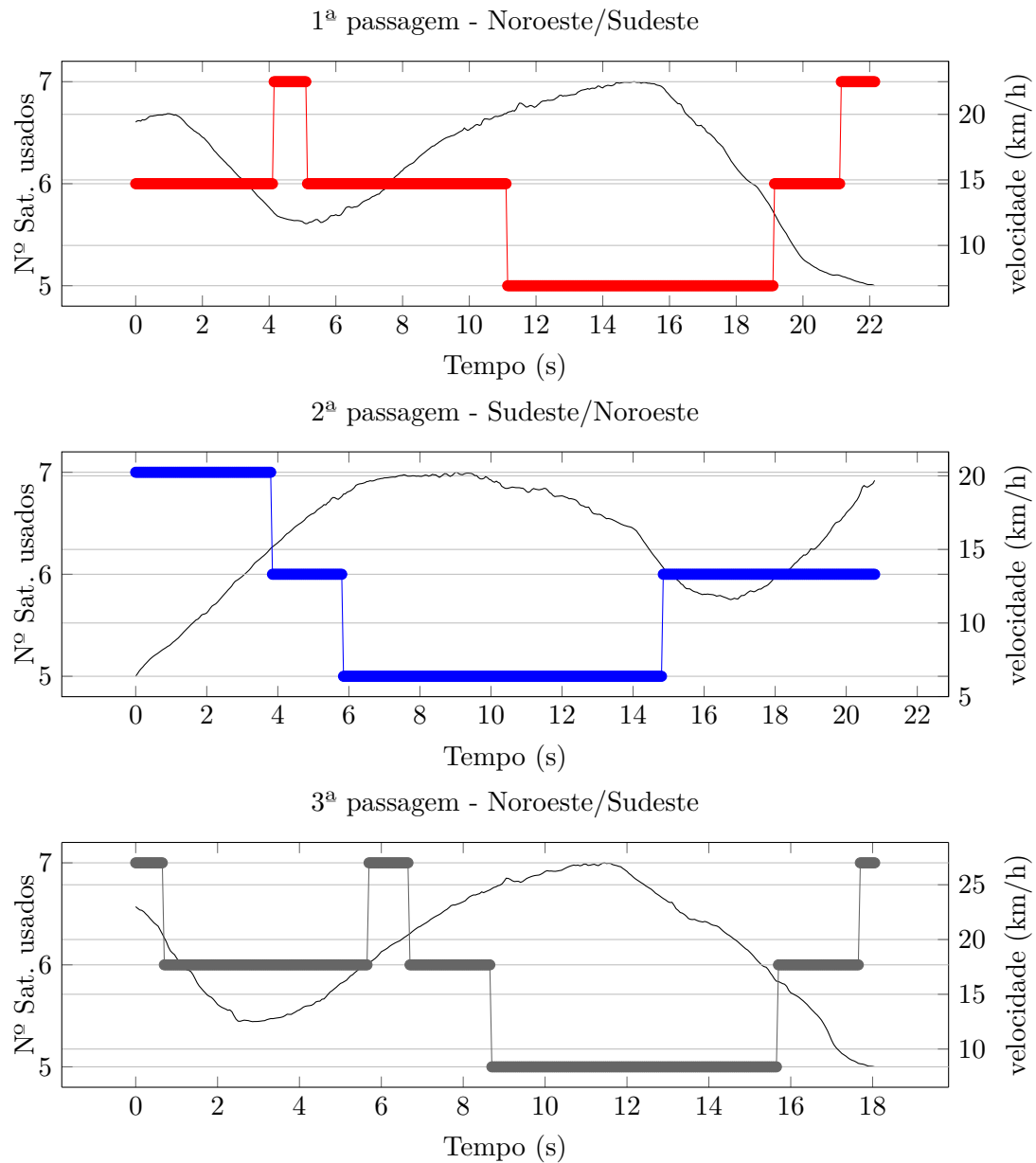


Figura 5.10: Número de satélites utilizados durante o período de tempo considerado na figura 5.9 e velocidade do veículo (curva de cor preta) durante o período considerado.

A figura 5.11 ilustra o valor de imprecisão durante o período alvo de estudo. Apesar de ocorrer uma subida considerável do valor de erro para valores próximos de 0.7 metros, este valor é considerado muito baixo tendo em conta as especificações do sensor. Contudo, como será demonstrado nas secções seguintes, se caso as condições adversas se prolongassem, isso levaria a que o valor de erro aumentasse sem limite definido, segundo a tendência de subida apresentada no gráfico.

Analisando a fase inicial das curvas da figura 5.11, sobressai a existência de uma

diferença muito significativa entre os valores de erro2D (equação 5.1). Essas diferenças podem ser explicadas pelo facto dos testes terem sido realizados pela ordem descrita anteriormente, sem que o sensor tivesse tempo para corrigir o erro acumulado pelas passagens anteriores.

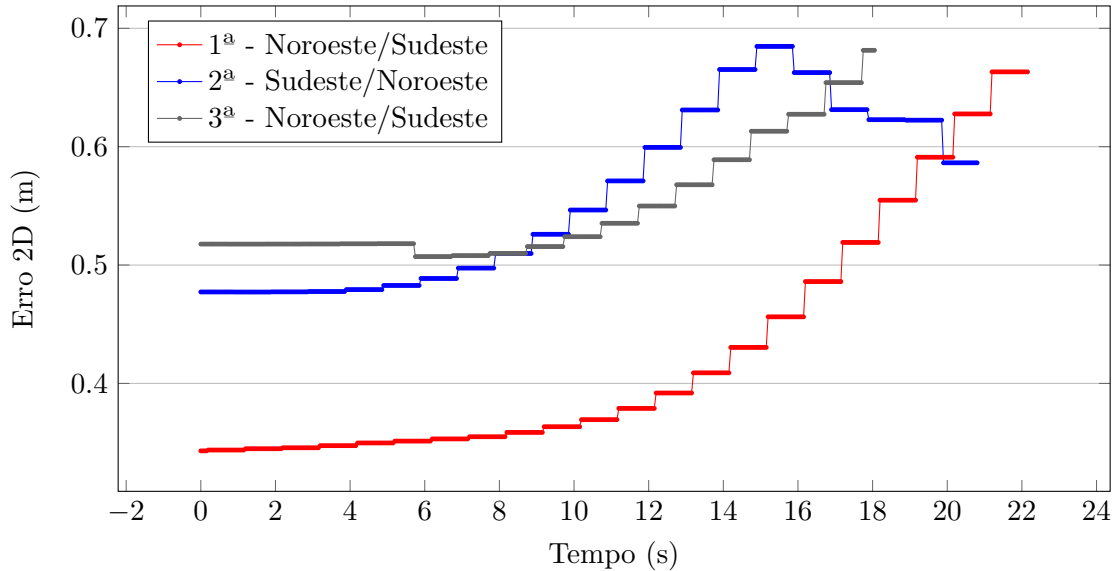


Figura 5.11: Valores de erro2D durante o período alvo de estudo, nas três passagens pela rua Prof. Manuel Est. Silva.

5.2.2 Túnel

A segunda situação escolhida para testar o desempenho do sensor foi a passagem por um túnel. Atualmente, o número de túneis, ou passagens subterrâneas no país aumentou consideravelmente, por isso é necessário estabelecer os limites de funcionamento do sistema nesse ambiente.

Na travessia de um túnel não existe a receção de sinais de satélite, sendo assim, o cálculo da posição é realizado apenas com base no sistema de navegação inercial. Desta forma, origina uma acumulação de erro que pode inviabilizar o normal funcionamento do veículo, ou seja, depois de um período de tempo a navegar apenas com base no sistema inercial o valor de posicionamento global pode indicar, por exemplo o posicionamento na via em sentido contrário o que poderá ter consequências na execução da missão.

Estes testes foram realizados em dois túneis. No túnel da Av. Santa Joana (Aveiro) com aproximadamente 210 metros de comprimento e tem um perfil em linha reta, foram realizadas 2 passagens em cada sentido. De modo a aumentar o nível de exigência foi realizada uma travessia do túnel entre a Av. Vasco Branco e a Av. Dr. Lourenço Peixinho (Aveiro) com 240 metros de comprimento e com a particularidade de ter um perfil em curva.

A figura 5.13 mostra os valores de posicionamento global durante a travessia do túnel. O intervalo de valores apresentado retrata o período de passagem no túnel e os momentos logo após a saída do túnel, de modo a mostrar o período de reaquisição de sinal dos satélites.

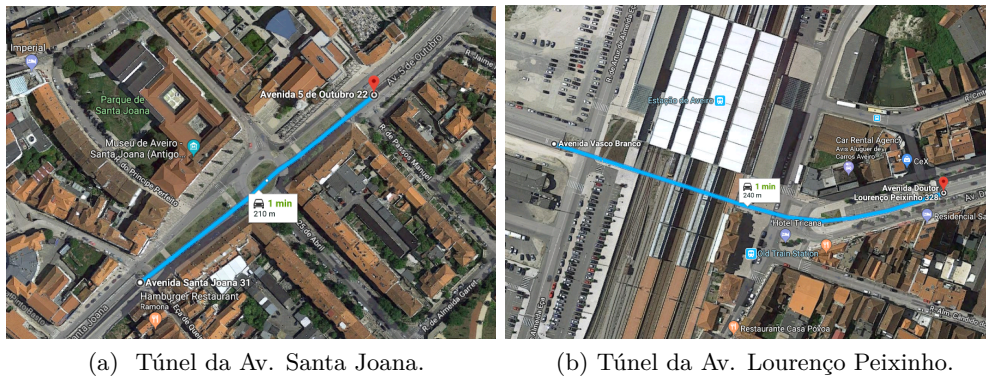


Figura 5.12: Perfis dos túneis utilizados para a realização dos testes.

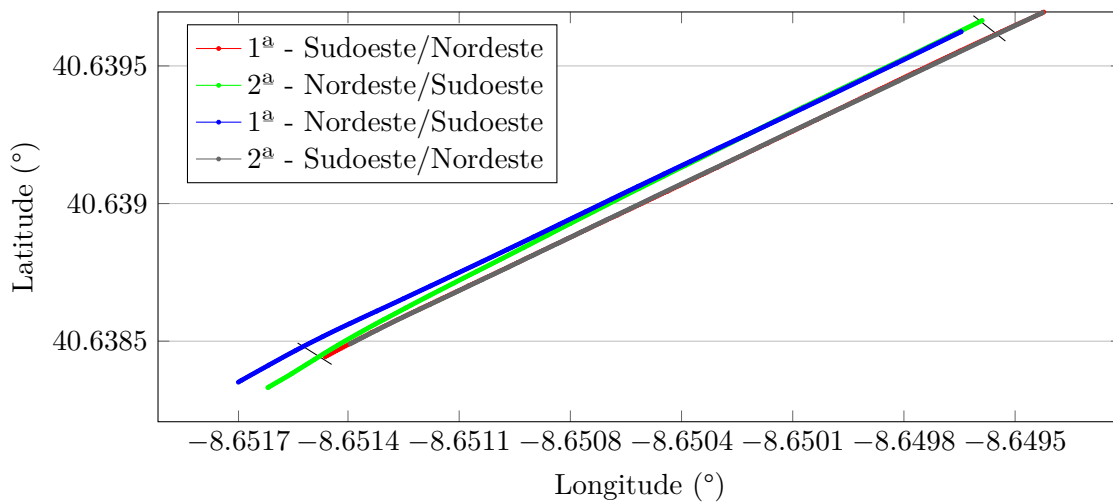


Figura 5.13: Repetibilidade das leituras de posicionamento global na passagem pelo túnel na Av. Santa Joana. Os segmentos de reta a preto definem os limites do túnel.

A figura 5.14 mostra o número de satélites utilizados pelo sensor no cálculo da posição. Como esperado, no interior do túnel não existe possibilidade de receção contínua de sinal, e apenas existe um pequeno período com a receção de sinal de um satélite, que corresponde à passagem numa zona em que o túnel tem um abertura para o exterior. Relativamente à velocidade do veículo, é esperado que tenha perfis distintos visto que um dos objetivos deste teste é estudar a influência do tempo sem sinal de satélites no valor de erro de posição, ou seja, como o túnel tem um comprimento fixo e pretende-se variar o tempo de atravessamento, conseqüentemente é necessário variar a velocidade.

Tal como já foi referido, de modo a aumentar o nível de dificuldade do desafio foi realizada uma passagem pelo túnel entre a Av. Vasco Branco e a Av. Dr. Lourenço Peixinho. Essa passagem foi realizada em 18,25 segundo e não foi usado nenhum satélite durante esse período.

A figura 5.15 mostra o valor do erro da posição global calculada durante o período de tempo estudado. Tanto as quatro curvas referentes às passagens pelo túnel da Av. Santa Joana como à passagem pelo túnel da Av. Lourenço Peixinho seguem a mesma

tendência, o que demonstra a excelente resposta da parte do sistema inercial.

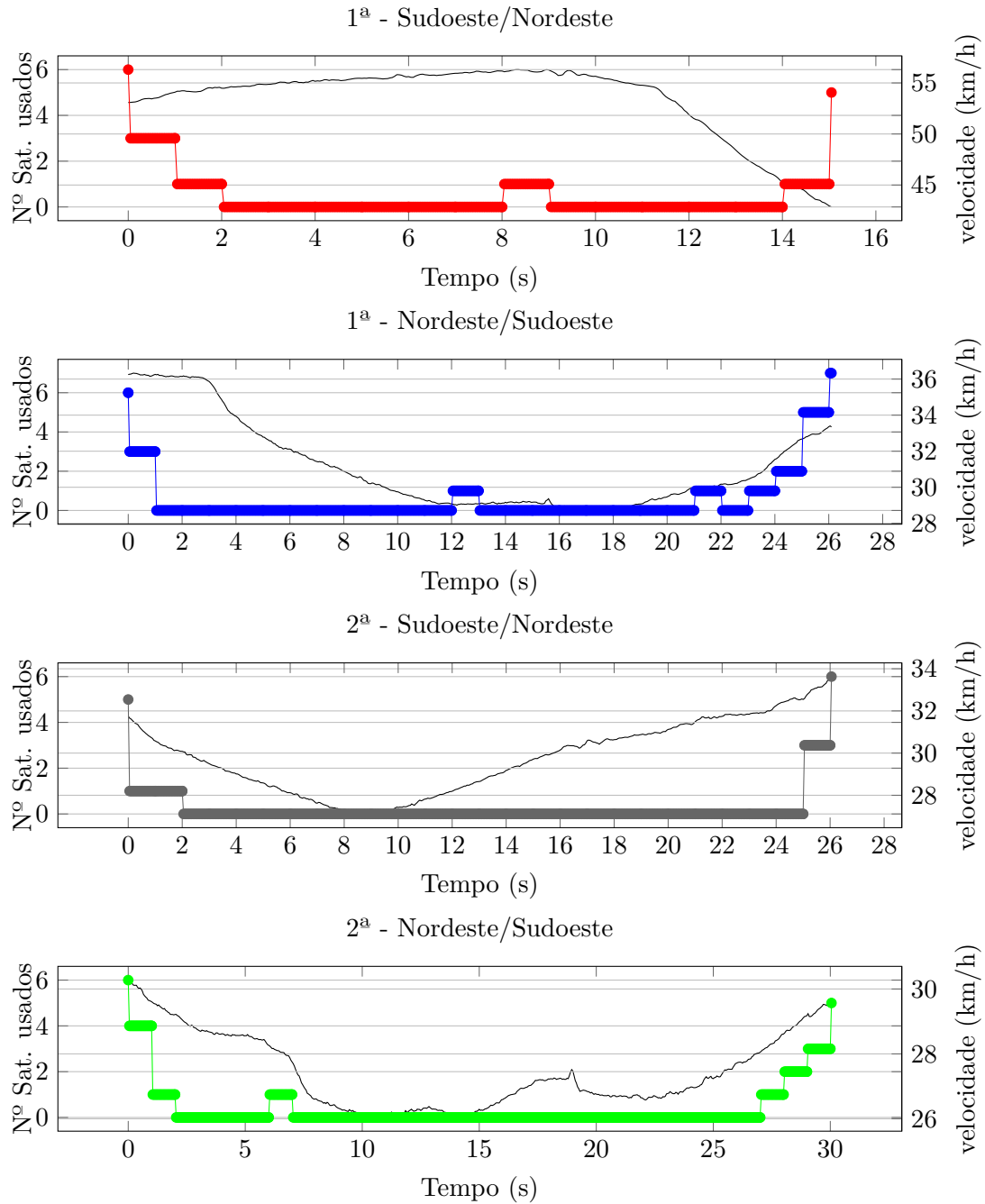


Figura 5.14: Representação do número de satélites utilizados na solução e velocidade do veículo durante o período correspondente ao deslocamento representado na figura 5.13. A curva preta traduz o valor de velocidade.

Nas figuras 5.16 e 5.17 é realizada a representação no mapa das leituras de posição global durante a realização dos testes. É de salientar que o período de tempo apresentado é apenas indicativo e os seus limites não estabelecem uma relação direta com os

períodos de tempo estudados. Assim, pretende-se avaliar se a incerteza das leituras tem consequências no nível de exatidão das mesmas. Apenas nas duas passagens realizadas no túnel da Av. Santa Joana no sentido Nordeste/Sudoeste, já depois da saída do túnel, existe um desfasamento entre a posição calculada e a posição real, em que o primeiro caso se deve a fenómenos de reflexão, enquanto no segundo caso deve-se a uma lomba no perfil da estrada que pode ter perturbado o sistema inercial.

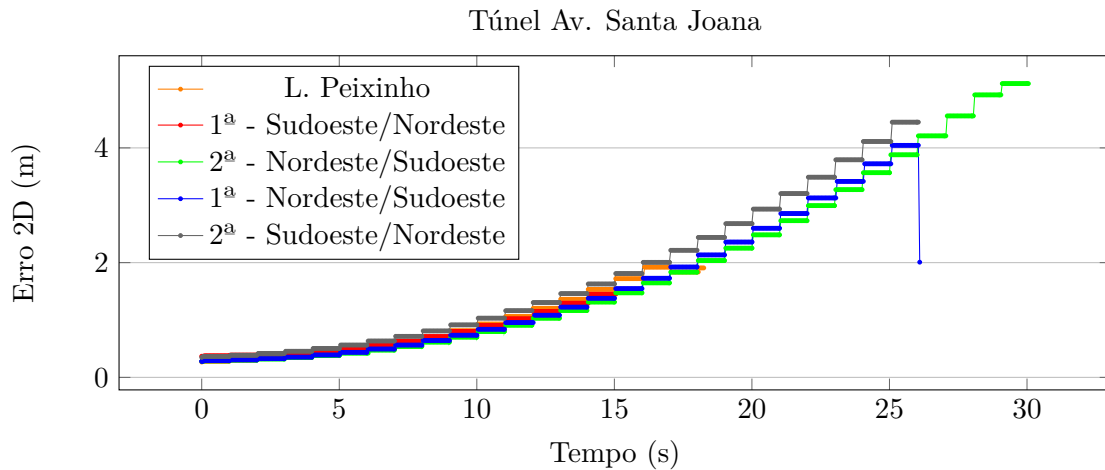


Figura 5.15: Valores de erro da posição global, calculados segundo a equação 5.1.

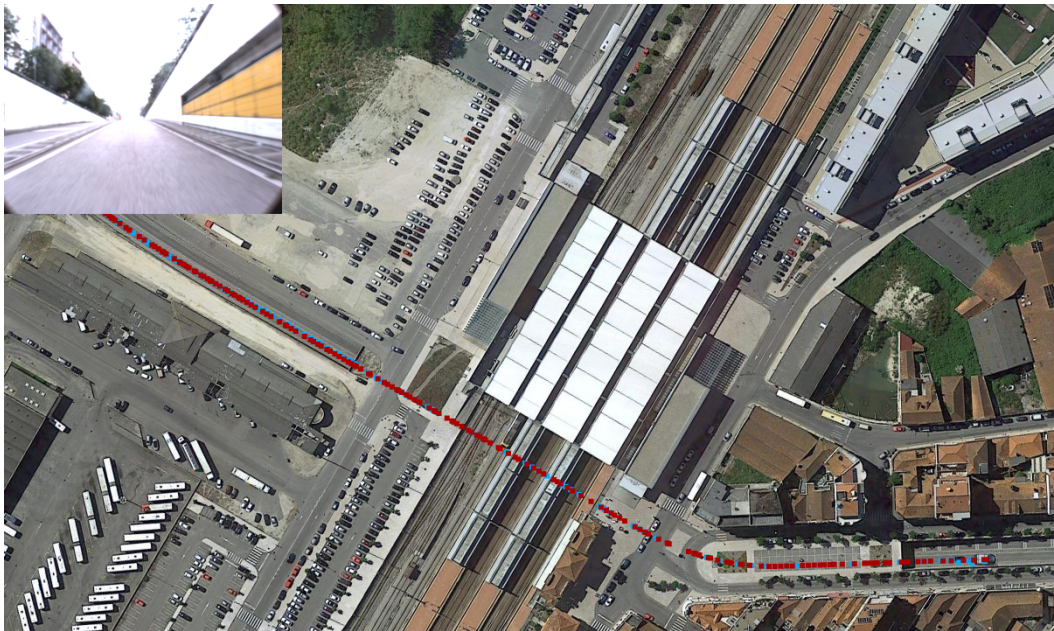


Figura 5.16: Representação no mapa, em vista satélite, das leituras de posicionamento global durante a passagem pelo túnel da Av. Lourenço Peixinho. A vermelho a posição usando a fusão GNSS+INS e azul apenas o sistema GNSS. Contudo, no interior do túnel ambas as cores representam a utilização do sistema inercial de modo a fornecer continuamente uma leitura ao utilizador.



(a) Primeira passagem no sentido Sudoeste/-Nordeste. (b) Primeira passagem no sentido Nordeste/Sudoeste.



(c) Segunda passagem no sentido Sudoeste/-Nordeste. (d) Segunda passagem no sentido Nordeste/Sudoeste.

Figura 5.17: Representação no mapa, em vista satélite, da posição global utilizando GNSS+INS (a vermelho) e da posição global apenas com base no sistema GNSS (a azul).

5.2.3 Rua com árvores de grande porte

Uma situação muito comum nas ruas urbanas e rurais é a existência de árvores de grande porte que condicionam a propagação dos sinais provocando fenômenos de reflexão. Para avaliar o desempenho do sistema GNSS+INS foram realizadas várias passagens na Avenida Artur Ravara, em Aveiro. Esta rua, para além de ser delimitada por árvores de grande porte, tem uma ponte pedonal e trata-se de uma rua empedrada o que constitui na globalidade um desafio muito interessante ao sistema.

Na figura 5.18 é possível observar os valores das leituras de posicionamento global fornecidas pelo sensor durante a realização dos testes. Foram realizadas duas passagens em cada sentido da avenida, de modo a obter uma amostra representativa das variáveis estudadas.

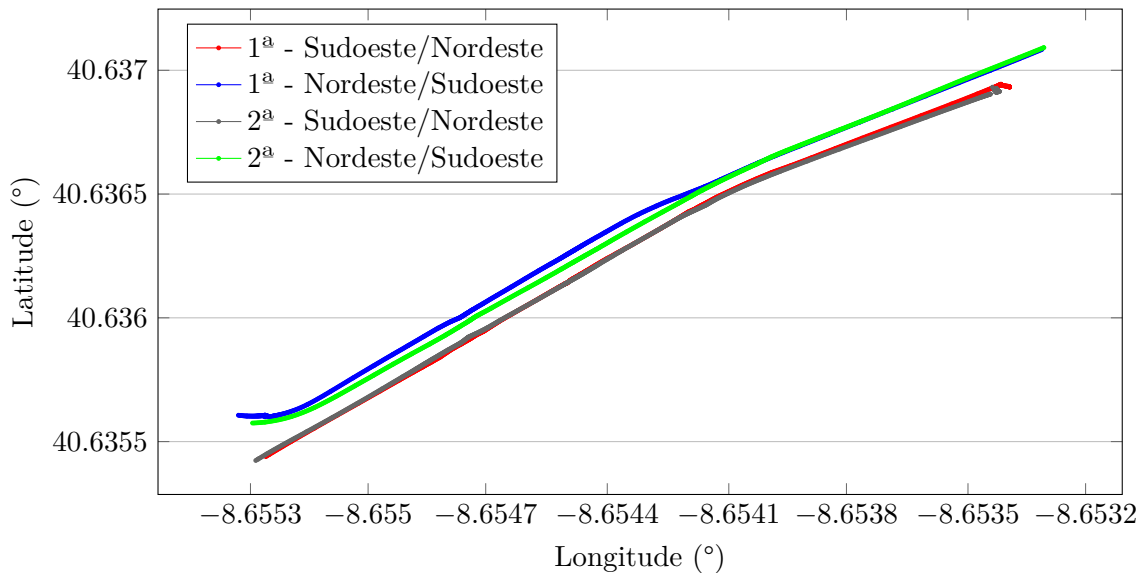


Figura 5.18: Repetibilidade das leituras de posicionamento global nas quatro passagens pela Av. Artur Ravara.

A figura 5.18 mostra um elevado nível de repetibilidade de leituras. No sentido sudoeste/nordeste apenas existe um ligeiro desfasamento no final da rua onde se encontram uns semáforos, e em ambas as passagens o veículo esteve parado devido ao sinal vermelho. Durante essa paragem ocorreu uma ligeira variação de posição devido à correção do valor de posição provocado pelo aumento do número de satélites utilizados.

Essa paragem originou umas ligeiras variações de posição, que são normais visto que o sistema tinha associado um erro elevado e com o aumento do número de satélites utilizados ocorre a convergência no cálculo da posição e consequentemente o valor de erro2D baixa consideravelmente (figura 5.20).

Quanto ao sentido nordeste/sudoeste existe um desfasamento considerável a partir de meio da avenida, contudo esse desfasamento está correto, pois a Avenida Artur Ravara possui duas vias e devido ao cumprimento do código da estrada na primeira passagem (curva azul) houve uma mudança para a via da direita.

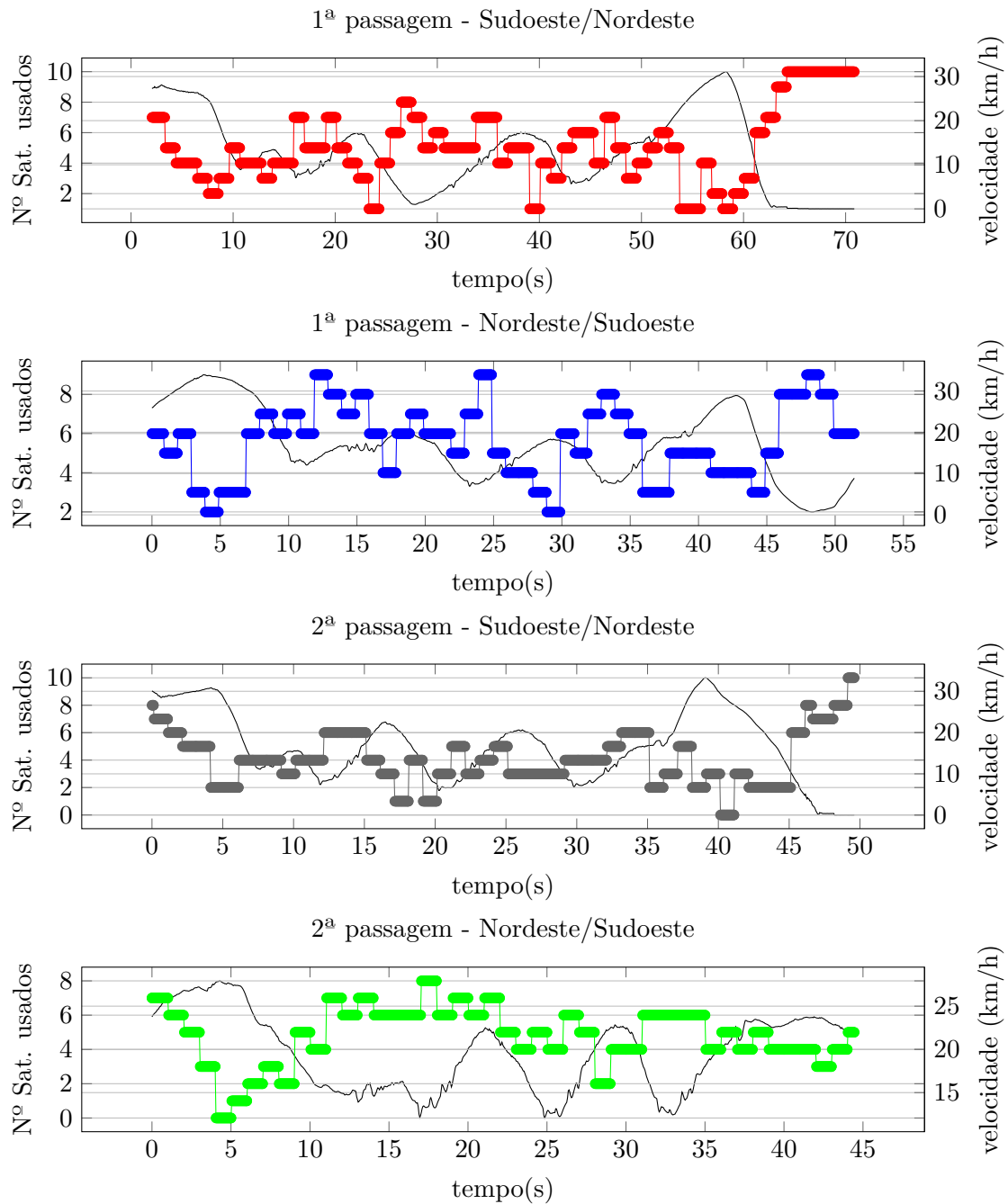


Figura 5.19: A colorido número de satélites utilizados pelo sensor no cálculo da posição e a preto o valor de velocidade do veículo nas quatro passagens pela Av Artur Ravara.

Os valores de erro obtidos durante estes testes estão diretamente relacionados com o número de satélites usados na solução; uma diminuição do número de satélites leva a um aumento do valor de erro. Pela análise das figuras 5.19 e 5.20, nomeadamente das curvas do sentido sudoeste/nordeste, depois de um período de aumento gradual do erro, a utilização de seis ou mais satélites origina uma diminuição do valor de erro

praticamente imediata.

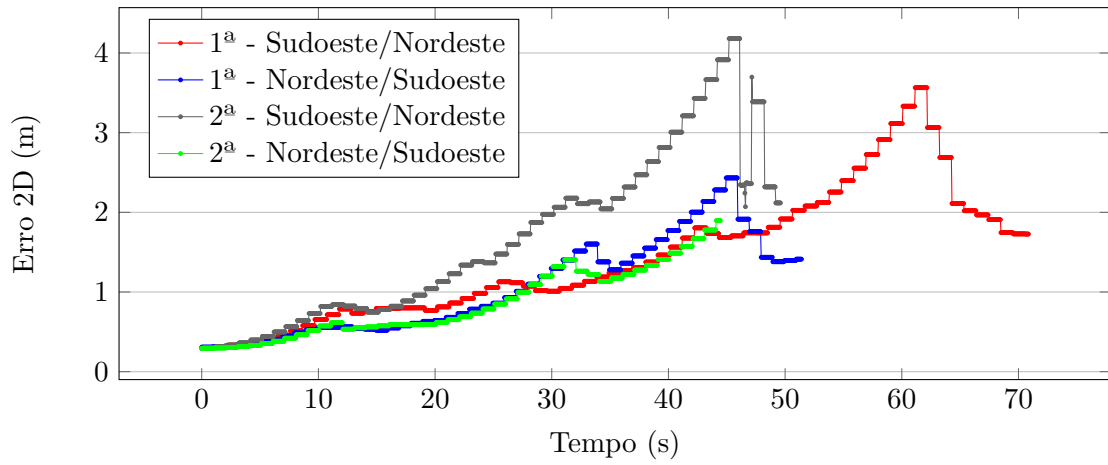
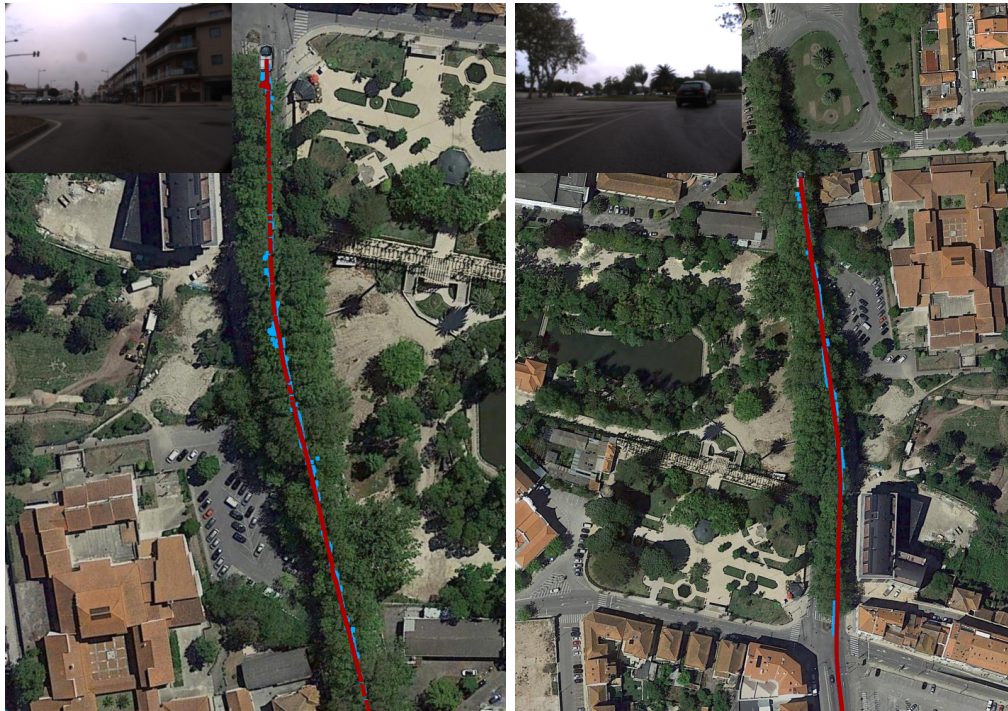


Figura 5.20: Valor de erro utilizando a equação 5.1 durante a passagem pela Av. Artur Ravara.

Na figura 5.21 pode-se observar o nível de desempenho da integração GNSS+ INS; apesar do baixo número de satélites usados e a elevada reflexão dos sinais originada pelas árvores e pela ponte pedonal, o posicionamento global não foi prejudicado. E mesmo nesta situação crítica, existe uma clara distinção entre o posicionamento em diferentes vias da estrada.



(a) Primeira passagem no sentido Sudoeste/Nordeste. (b) Primeira passagem no sentido Nordeste/Sudoeste.



(c) Segunda passagem no sentido Sudoeste/-Nordeste. (d) Segunda passagem no sentido Nordeste/Sudoeste.

Figura 5.21: Representação no mapa em vista satélite da posição global utilizando GNSS+INS (a vermelho) e da posição global apenas com base no sistema GNSS (a azul) .

5.3 Planeamento e execução de rota

Depois de avaliado o desempenho do sistema de localização global, é necessário analisar o funcionamento do algoritmo de planeamento e execução de rota.

De modo a estudar o funcionamento da metodologia de planeamento, foi estabelecida uma rota que passa por curvas apertadas, cruzamentos e uma rotunda. A figura 5.22 mostra o trajeto entre os dois pontos em que será analisada a evolução do índice do segmento de referência e a distância ao *waypoint* que define o final do segmento de referência.

Na figura 5.23 são apresentados o valor do índice do segmento de referência durante a execução da rota representada na figura 5.22 e o valor da distância entre a posição absoluta do veículo e o ponto W_{i+1} (secção 4.3). Pela análise do gráfico pode-se verificar que não foram ultrapassados segmentos de caminho, e quando a distância entre a posição do veículo e ponto final do segmento é inferior a 4 metros ocorre a transição de segmento e, conseqüentemente, a distância ao ponto seguinte aumenta.



Figura 5.22: Rota entre A e B usada no estudo do funcionamento do algoritmo de planeamento desenvolvido.

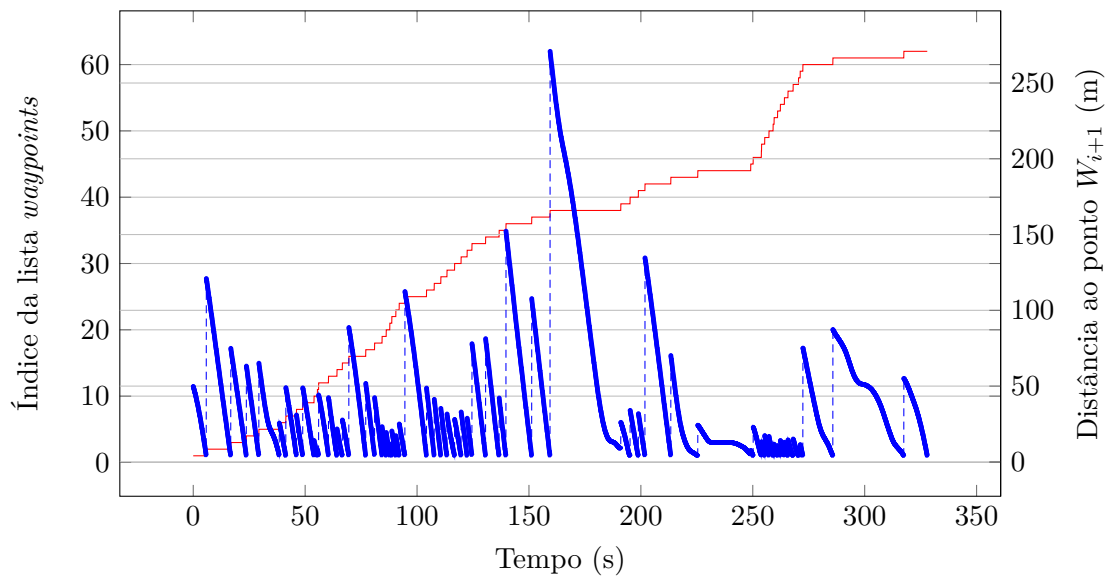


Figura 5.23: Valor de S_i (a vermelho) e distância ao *waypoint* seguinte (a azul) em função do tempo.

Como descrito na secção 4.3, a primeira fase do planeamento global inicia-se com o pedido do utilizador que provoca uma requisição de rota através do serviço *Google*

Maps API Directions. O sucesso do algoritmo de planeamento e execução de rota passa pela utilização de serviço externo que cumpra os requisitos estabelecidos pelo código da estrada. Por isso, na secção 5.3.1 será apresentada uma situação ilustrativa do cumprimento das regras de trânsito que qualquer veículo deve seguir.

A segunda fase do processo consiste em encontrar o segmento de referência que o módulo de navegação local deve utilizar para chegar a um destino e proceder à verificação do cumprimento da rota e, em casos de saída de rota, proceder a um novo pedido. Na secção 5.3.2 será mostrada a resposta do planeador global numa situação de saída de rota.

5.3.1 Aplicação do código da estrada

Todos os veículos que circulam nas vias públicas têm de respeitar as regras impostas pela legislação. O código da estrada impõe diversas restrições, como por exemplo, respeitar a sinalização colocada na via. A rota fornecida pelo serviço externo deve ter em conta a sinalização presente no local; por exemplo, as ruas de sentido único só podem ser utilizadas num só sentido e essa regra deve ser refletida na rota fornecida pelo serviço externo.

De modo a aferir a compatibilidade das rotas fornecidas pelo serviço externo com as regras do código da estrada, nomeadamente, o cumprimento do sentido de deslocação em ruas com apenas um sentido, foi estabelecido um destino numa rua de sentido único, e que dada a posição do veículo o trajeto mais curto seria a deslocação em contra-mão.

Na figura 5.24 é possível observar a execução da missão cujo o destino é representado pela letra *B*, e a rota fornecida pelo serviço está representada a preto, e os dois *waypoints* a amarelo definem o segmento de referência naquele momento.

Como se pode visualizar, a rota fornecida tem em conta o sentido de deslocamento permitido em cada via. Este fator permite ter um elevado nível de confiança no serviço utilizado, contudo, numa aplicação de elevado grau de complexidade, como é o caso dos veículos autónomos, é aconselhável estabelecer processos redundantes de modo a garantir que a rota esteja de acordo com o código da estrada.

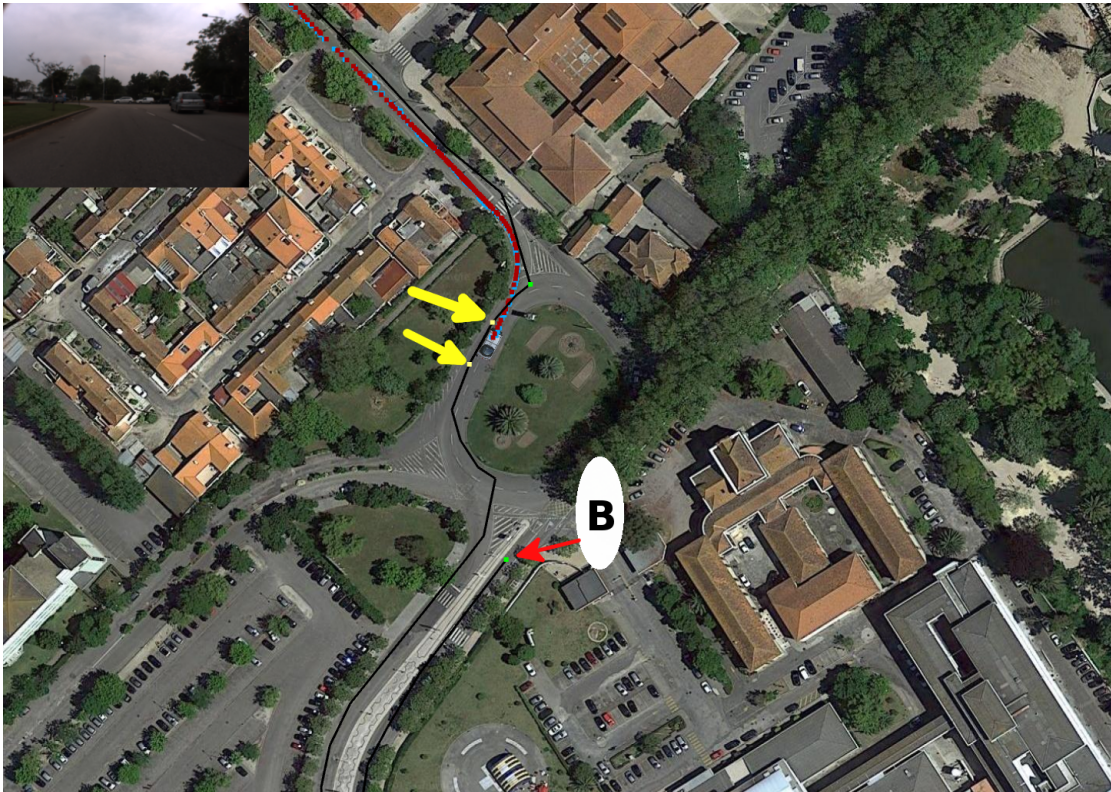


Figura 5.24: Demonstração do planeamento numa situação em que o destino (B) da rota se encontra numa via de sentido único. Os pontos a amarelo definem o segmento de referência.

5.3.2 Saída de rota

Uma das características mais relevantes da metodologia de planeamento desenvolvida, é a capacidade de detetar se o veículo saiu de rota e proceder a um novo pedido de rota, de modo a encontrar um novo caminho para o destino introduzido pelo utilizador.

Na figura 5.25 está representada uma rota entre dois locais que o veículo não respeitou. Neste caso, o veículo optou pelo uso da via da esquerda, que o conduziu a um túnel que não fazia parte da rota planeada. Devido à pequena distância entre o túnel e a rota fornecida, só no momento de passagem pela rotunda o planeador detetou a saída de rota e procedeu a um novo pedido de rota.

Na figura 5.26 é mostrada a nova rota entre a posição em que o planeador detetou a saída de rota e o local de destino previamente estabelecido. A linha reta é meramente indicativa, pois une o local de deteção da saída de rota ao local de destino.

A utilização de uma distância inferior a 10 metros como limiar da saída de rota poderia ter permitido a deteção da saída de rota mais cedo, contudo, dado que diminuição desse valor poderia provocar o uso desmesurado do serviço externo e por outro lado a deteção mais cedo não evitaria a saída de rota, portanto o valor de 10 metros permite flexibilizar as trajetórias do veículo face à rota, mas é eficaz no seu propósito.



Figura 5.25: Execução de rota entre o ponto *A* e *B*.



Figura 5.26: Recalculo da rota entre o ponto *C* e *B* depois de detetada a saída de rota.

Nas figuras 5.27 e 5.28 é apresentado outro exemplo de saída de rota. Neste exemplo, o veículo não respeitou a rota estabelecida ao utilizar uma saída errada da rotunda.

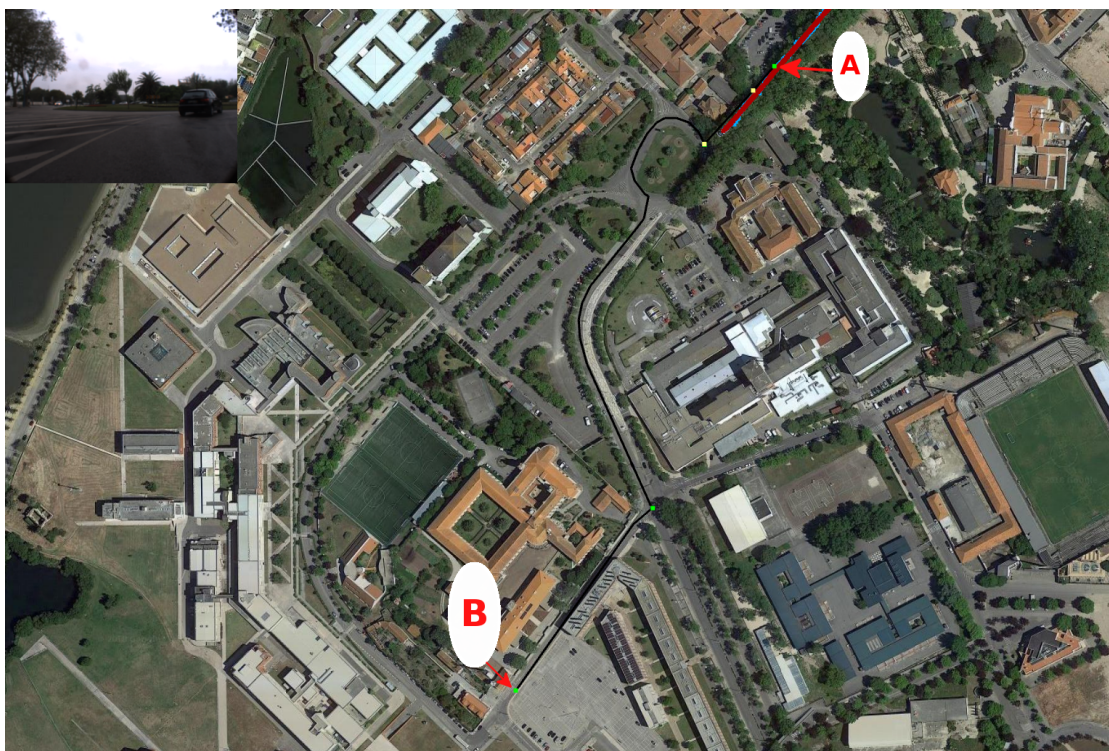


Figura 5.27: Execução de rota entre o ponto A e B.

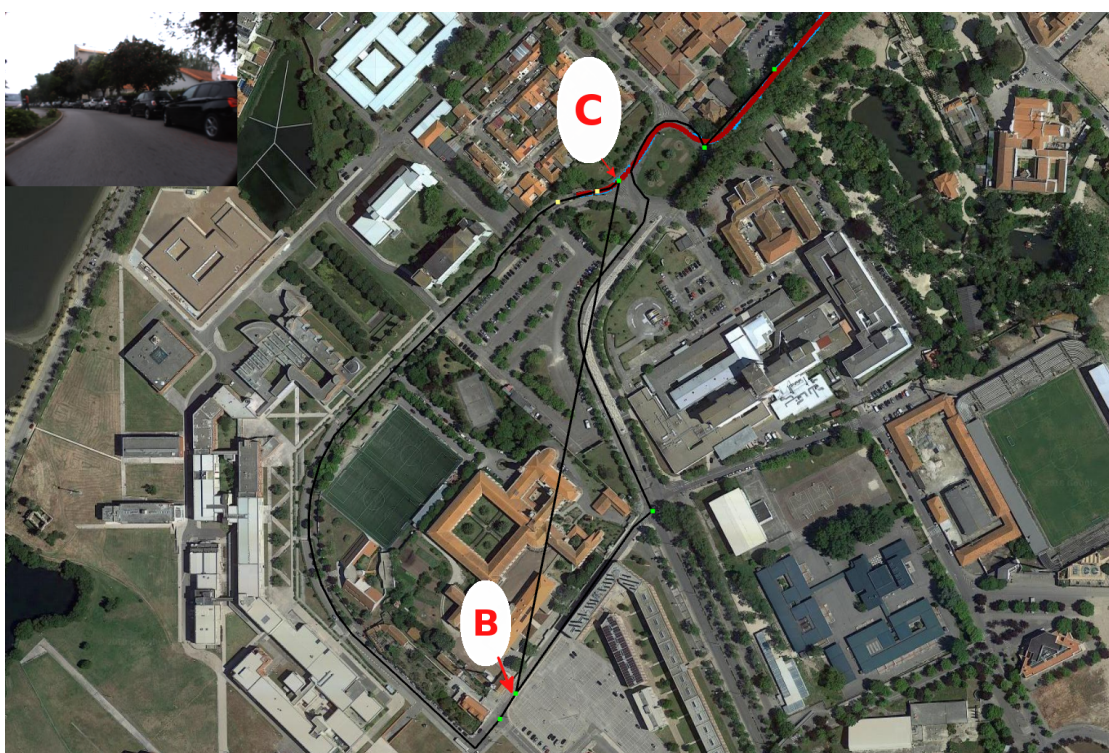


Figura 5.28: Recalculo da rota entre o ponto C e B depois de detetada a saída de rota.

Capítulo 6

Conclusões e Trabalho Futuro

6.1 Conclusões

Esta dissertação teve como objetivo fornecer uma solução de localização e navegação global para o ATLASCAR 2. A solução desenvolvida tem como base a utilização do sensor GNSS+INS, que garante uma leitura de posicionamento global de elevado rigor. No plano da navegação, sobressai a utilização do serviço *on-line Google Maps API Directions* como fonte de rotas. Por fim, na perspectiva de monitorização do estado do veículo, são desenvolvidas duas bases de dados em *Postgresql*.

A localização global representa os alicerces de todo o trabalho, e a correta instalação e configuração do veículo permitiu alcançar leituras de posicionamento global no plano com incerteza inferior a um metro. Em situações favoráveis, isto é, em situações de receção de sinais de um elevado número de satélites e a não ocorrência de fenómenos de reflexão, a incerteza alcançada é bastante inferior a um metro. Os testes apresentados na secção 5.2 mostram a resposta do sistema em situações extremas onde a precisão das leituras de posicionamento se degrada, em alguns casos, largamente para além da barreira de um metro, contudo, a posição global continua com um elevado grau de exatidão. A representação destas leituras no mapa comprova o nível de exatidão da posição fornecida pelo sistema.

A metodologia de navegação global implementada pretende cumprir a tarefa de fornecer ao módulo de navegação local segmentos de caminho que conduzam o veículo até ao destino. Esta metodologia apresenta-se como uma solução flexível, que fornece indicação sobre a direção que o veículo deve seguir e não tem como objetivo estabelecer uma trajetória rígida. O algoritmo baseado em distâncias euclidianas revelou-se, durante os testes, bastante eficaz nas situações de saída de rota, e tanto em casos de segmentos longos ou curtos, o algoritmo respeitou sempre a sequência de segmentos de caminho definidos pelos *waypoints*.

A utilização da aplicação *Mapviz* permitiu a visualização de variáveis relativas ao posicionamento global e muitos outros dados, como a visualização de nuvens de pontos ou imagens de uma câmara. Para além dessas características, foram desenvolvidos novos *plugins* de modo a que a aplicação funcione como interface com o utilizador no momento da introdução do destino, e seja possível a representação da rota e parâmetros relativos ao planeamento global como, por exemplo, os *waypoints* que definem o segmento de referência.

Por fim, a tarefa de monitorização remota realizada através de uma base de dados

Postgresql, desenvolvida de raiz, permitiu colocar em funcionamento a página *web*. Deste modo, é possível aceder de qualquer local à posição do veículo e outras variáveis como, por exemplo, a velocidade instantânea e o destino da missão. A base de dados local, encontra-se em funcionamento apenas com informação relativa ao posicionamento e navegação global.

6.2 Trabalho futuro

Este trabalho propõe uma solução que resolve as necessidades relativas ao posicionamento e navegação global do veículo ATLASCAR 2. Contudo, existem alguns trabalhos que podem ser realizados de modo a potencializar a solução proposta, tanto no plano da localização como no processo de navegação global:

- **Localização global:** - A realização deste trabalho abre portas para que no futuro seja instalado um hodómetro no veículo ou, por exemplo, seja simulada a pulsação de hodómetro decorrente da informação proveniente da centralina do veículo, e esta seja introduzida no sensor através das portas dedicadas para esse efeito. A instalação de um hodómetro permitiria baixar o erro das leituras de posição em situações de receção de sinal de um baixo número de satélites.
- **Navegação global:** - Na área do planeamento e execução de rota é necessário criar um método de utilização das instruções de navegação como, por exemplo, "vire à direita", para realizar o acionamento dos piscas de acordo com a mudança de direção a realizar e adequar a velocidade do veículo à situação. Esta informação poderá ser fornecida ao planeador local de modo a limitar a direção das trajetórias segundo a instrução de navegação. Este trabalho permitirá evitar a utilização de *waypoints* mal posicionados em entroncamentos ou rotundas. A figura 6.1 mostra uma situação em que os *waypoints*, que indicam a mudança de direção, estão posicionados incorretamente.



Figura 6.1: Situação e posicionamento errado dos *waypoints*.

Referências

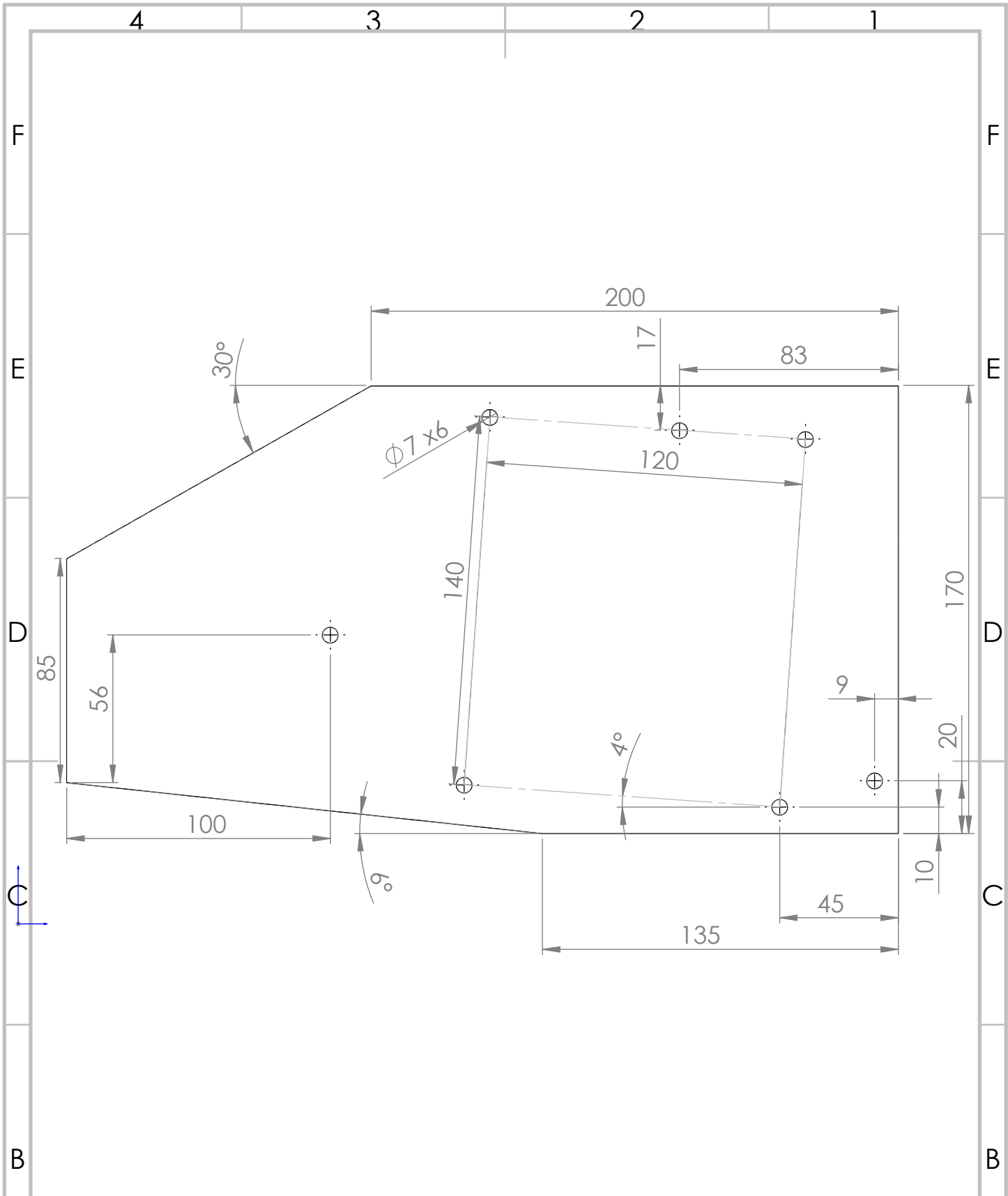
- Google (2018). *Google Maps Platform*. URL: <https://cloud.google.com/maps-platform/> (acedido em 16/05/2018).
- NovAtel (2018). *High Performance GNSS GPS Antennas*. URL: <https://www.novatel.com/products/gnss-antennas/high-performance-gnss-antennas/gps-702-gg/%7B%5C%7DalternateProducts> (acedido em 15/05/2018).
- Service (2018). *Ros wiki*. URL: <http://wiki.ros.org/Services> (acedido em 17/05/2018).
- Topics (2018). *ROS wiki*. URL: <http://wiki.ros.org/Topics> (acedido em 17/05/2018).
- Correia, José (2017). “Unidade de Percepção Visual e de profundidade para o ATLAS-CAR2”. Dissertação de Mestrado. Universidade de Aveiro, pp. 1–98.
- Cui, Bingbo, Xiyuan Chen e Xinhua Tang (2017). “Improved Cubature Kalman Filter for GNSS/INS Based on Transformation of Posterior Sigma-Points Error”. Em: 65.11, pp. 2975–2987.
- Novatel (2017). *SPAN*. Rel. téc., p. 12. URL: <https://www.novatel.com/assets/Documents/Papers/SPANBrochure.pdf>.
- NovAtel Inc. (2017). “OEM6 Firmware Reference Manual”. Em: January, p. 756. URL: <https://www.novatel.com/assets/Documents/Manuals/om-20000129.pdf>.
- Roysdon, Paul F. e Jay A. Farrell (2017). “Robust GPS-INS Outlier Accommodation: A Soft-thresholded Optimal Estimator”. Em: *IFAC-PapersOnLine* 50.1, pp. 3574–3579. ISSN: 24058963. DOI: 10.1016/j.ifacol.2017.08.682. URL: <https://doi.org/10.1016/j.ifacol.2017.08.682>.
- Snider, Daniel (2017). *ROS Offline Google Maps for MapViz*. URL: <https://github.com/danielsnider/MapViz-Tile-Map-Google-Maps-Satellite> (acedido em 15/05/2018).
- Teunissen, Peter J.G. e Oliver Montenbruck (2017). *Springer Handbook of Global Navigation Satellite Systems*. Springer International Publishing, p. 1327. ISBN: 9783319429267. DOI: 10.1007/978-3-319-42928-1.
- Wu, Garin e Xinzhou (2017). “ROBUST POSITIONING FROM VISUAL-INERTIAL AND GPS”. Em: *North Coast Media LLC*, p. 72.
- NovAtel (2016). *SPAN-IGM-A1 datasheet*. Rel. téc., p. 2. URL: <https://www.novatel.com/products/gnss-antennas/high-performance-gnss-antennas/gps-702-gg/>.
- NovAtel Inc. (2016). “SPAN on OEM6 Firmware Reference Manual”. Em: December. URL: <https://www.novatel.com/assets/Documents/Manuals/OM-20000144UM.pdf>.
- Paden, Brian, Michal Cap, Sze Zheng Yong, Dmitry Yershov e Emilio Frazzoli (2016). “A Survey of Motion Planning and Control Techniques for Self-driving Urban Vehicles”.

- Em: 1.1, pp. 33–55. ISSN: 2379-8904. DOI: 10.1109/TIV.2016.2578706. arXiv: 1604.07446. URL: <http://arxiv.org/abs/1604.07446>.
- Fernandez, Enrique, Luis Sanchez Crespo, Anil Mahtani e Aaron Martinez (2015). *Learning ROS for Robotics Programming*. Second Edi. Packt Publishing, p. 415. ISBN: 9781783987580.
- Ferreira, Ivan (2015). “Integration of the planning, global localization and path execution in autonomous cars”. Dissertação de Mestrado. Universidade de Aveiro.
- Lemos, Randerson Araujo De, Olmer Garcia e Janito Vaqueiro Ferreira (2015). “Local and Global Path Generation for Autonomous Vehicles Using Splines - Extended Version”. Em: pp. 1–13. DOI: 10.14483/udistrital.jour.reving.20XX.X.aXX.
- NovAtel (2015a). *An Introduction to GNSS*. second edi. NovAtel Inc., p. 91. ISBN: 9780981375403.
- (2015b). *Gps-701-GG & Gps-702-GG*. Rel. téc., p. 2. URL: https://www.novatel.com/assets/Documents/Papers/GPS701%7B%5C_%7D702GG.pdf.
- (2015c). “Precise Positioning with NovAtel CORRECT Including Performance Analysis”.
- Won, Dae Hee, Eunsung Lee, Moonbeom Heo, Seung-woo Lee, Jiyun Lee, Jeongrae Kim, Sangkyung Sung e Young Jae Lee (2014). “Selective Integration of GNSS, Vision Sensor, and INS Using Weighted DOP Under GNSS-Challenged Environments”. Em: 63.9, pp. 2288–2298.
- Houghton, Joshua e Sandy Kennedy (2013). “Analysis of a tightly - coupled mems imu based gnss/ins with an external odometer”. Em: pp. 1–13.
- NovAtel Inc. (2013). *SPAN-IGM-A1™ User Manual*.
- Viana, José (2013). “E-Stop e consola de controlo e monitorização para o AtlasCar”. Dissertação de Mestrado. Universidade de Aveiro.
- ZENEC (2013). *Gps navigation system nav user manual motor home*. Rel. téc. URL: http://www.zenec.com/wp-content/uploads/user%7B%5C_%7Dupload/Z-N720MH/Operating%7B%5C_%7DNavigation/ZENEC%7B%5C_%7DNavUserManual-MHS%7B%5C_%7DV201%7B%5C_%7DEN.pdf.
- NovAtel (2009). *GPS-702-GG, GPS-701-GG and GPS-702-GG-N • • USER GUIDE*. Rel. téc., p. 2.
- Min, Hyun-Suk (2006). *Navigation System and Method For Deteting Deviation of Mobile Objects From Route Using Same*. URL: <https://patentimages.storage.googleapis.com/4b/22/a1/96f3771f5aba9b/US7653484.pdf>.
- Katayama, Mutsumi e Hiroyuki Morita (2002). *Mobile Navigation Apparatus with route deviation indication*. URL: <https://patentimages.storage.googleapis.com/c2/5e/5a/f26b7f8fc4e129/US6487496.pdf>.
- McKerrow, P. (1991). *Introduction to Robotics*. Addison-Wesley Publishing Company, p. 811.

Apêndices

Apêndice A

Suporte



UNLESS OTHERWISE SPECIFIED:
 DIMENSIONS ARE IN MILLIMETERS
 SURFACE FINISH:
 TOLERANCES:
 LINEAR:
 ANGULAR:

FINISH:

DEBURR AND
 BREAK SHARP
 EDGES

DO NOT SCALE DRAWING

REVISION

	NAME	SIGNATURE	DATE
DRAWN	Pedro		12-03-2018
CHK'D			
APPV'D			
MFG			

TITLE:
ISO 2768-M

Q.A.		MATERIAL: Chapa Alumínio 2mm
SOLIDWORKS Educational Product. For Instructional Use Only		WEIGHT:

DWG NO.
Suporte

SCALE:2:1

SHEET 1 OF 1

Apêndice B

Google Maps API Directions

B.1 Exemplo de mensagens

Mensagem de pedido de rota.

```
https://maps.googleapis.com/maps/api/directions/json?origin
=40.637543,-8.658705&destination=40.635677,-8.654912&key=****
```

Mensagem recebida em Json.

```
{
  "geocoded_waypoints" : [
    {
      "geocoder_status" : "OK",
      "place_id" : "ChIJ8eNlvwGYIwORBA6yPAjGOB8",
      "types" : [ "street_address" ]
    },
    {
      "geocoder_status" : "OK",
      "place_id" : "ChIJw4TTLQeYIwORiCWRkIbt018",
      "types" : [ "route" ]
    }
  ],
  "routes" : [
    {
      "bounds" : {
        "northeast" : {
          "lat" : 40.6378838,
          "lng" : -8.654959099999999
        },
        "southwest" : {
          "lat" : 40.6351735,
          "lng" : -8.6587364
        }
      },
      "copyrights" : "Map data 2018 Google",
      "legs" : [
        {
          "distance" : {
            "text" : "0.6 km",
            "value" : 555
          },
          "duration" : {
            "text" : "1 min",
            "value" : 88
          },
          "end_address" : "Av. Artur Ravara, 3810-164 Aveiro, Portugal",
          "end_location" : {
```

```

    "lat" : 40.6357075,
    "lng" : -8.654959099999999
  },
  "start_address" : "R. da Pega 19, 3810-164 Aveiro, Portugal",
  "start_location" : {
    "lat" : 40.6375552,
    "lng" : -8.6587364
  },
  "steps" : [
    {
      "distance" : {
        "text" : "41 m",
        "value" : 41
      },
      "duration" : {
        "text" : "1 min",
        "value" : 4
      },
      "end_location" : {
        "lat" : 40.6378838,
        "lng" : -8.6585249
      },
      "html_instructions" : "Head \u003cb\u003enorth-east\u003c/b\u003e on \u003cb\u003eR. da Pega\u003c/b\u003e towards \u003cb\u003eRua de Calouste Gulbenkian\u003c/b\u003e/\u003cb\u003eN235\u003c/b\u003e",
      "polyline" : {
        "points" : "w_`wFbdzs@_Ak@"
      },
      "start_location" : {
        "lat" : 40.6375552,
        "lng" : -8.6587364
      },
      "travel_mode" : "DRIVING"
    },
    {
      "distance" : {
        "text" : "0.3 km",
        "value" : 345
      },
      "duration" : {
        "text" : "1 min",
        "value" : 47
      },
      "end_location" : {
        "lat" : 40.6357095,
        "lng" : -8.655678999999999
      },
      "html_instructions" : "Turn \u003cb\u003eright\u003c/b\u003e onto \u003cb\u003eRua de Calouste Gulbenkian\u003c/b\u003e/\u003cb\u003eN235\u003c/b\u003e",
      "maneuver" : "turn-right",
      "polyline" : {
        "points" : "wa`wFvbzs@FUFQFOBId@_Af@q@pC_EVYZg@RYFI\\c@Ve@JEb@M"
      },
      "start_location" : {
        "lat" : 40.6378838,
        "lng" : -8.6585249
      },
      "travel_mode" : "DRIVING"
    },
    {
      "distance" : {
        "text" : "0.2 km",
        "value" : 169
      },
      "duration" : {
        "text" : "1 min",

```

```

        "value" : 37
    },
    "end_location" : {
        "lat" : 40.6357075,
        "lng" : -8.654959099999999
    },
    "html_instructions" : "At the roundabout, take the \u003cb\u003e3rd\u003c/b\u003e exit onto \u003cb\u003eAv. Artur Ravara\u003c/b\u003e\u003cdiv style =\u003cfont-size:0.9em\u003c\u003eDestination will be on the right\u003c/div\u003e",
    "maneuver" : "roundabout-right",
    "polyline" : {
        "points" : "et_wF~pys@DHDFFFHFJFHDLDDBAB?BABABA@ABCBC@A?A@A?A@??A?A@E@E?E@E?C?EAE?EAGAGCGCECEEECECEAIAa@k@UW"
    },
    "start_location" : {
        "lat" : 40.6357095,
        "lng" : -8.655678999999999
    },
    "travel_mode" : "DRIVING"
}
],
"traffic_speed_entry" : [],
"via_waypoint" : []
}
],
"overview_polyline" : {
    "points" : "w_`wFbdzs@_Ak@FUNa@h@iAxDqFnAeBt@iAn@SJPPNTLPBNEHIFQBUCYM[WOIAa@k@UW"
},
"summary" : "Rua de Calouste Gulbenkian/N235",
"warnings" : [],
"waypoint_order" : []
}
],
"status" : "OK"
}

```

B.2 Função de decodificação da *polyline*

A função seguinte, desenvolvida por Ferreira, 2015 é usada na decodificação da *polyline*. Contudo, esta função só funciona corretamente, se a *polyline* não contiver os caracteres `//`, neste caso uma das barras deve ser eliminada.

```

vector<double> Decompress(string encodedPoints, int precision)
{
    int len = encodedPoints.length();
    int index = 0;
    double lat = 0;
    double lng = 0;
    vector<double> array;

    while (index < len)
    {
        int b;
        int shift = 0;
        int result = 0;
        do
        {
            b = (int)encodedPoints[index++] - 63; // gets ascii value of the char
            // result |= (b & 0x1f) << shift;
            result |= (b & 31) << shift;
            shift += 5;
        } while (b >= 32);
        int dlat = ((result & 1) ? -(result >> 1) : (result >> 1));
        lat += dlat;
    }
}

```

```
shift = 0;
result = 0;

do
{
    b = (int)encodedPoints[index++] - 63; // gets ascii value of the char
    result |= (b & 31) << shift;
    shift += 5;
} while (b >= 32);
int dlng = ((result & 1) ? ~(result >> 1) : (result >> 1));
lng += dlng;

array.push_back(lat * pow(10, -precision));
array.push_back(lng * pow(10, -precision));
}
return array;
}
```

Apêndice C

Comunicação com o sensor

C.1 Implementação da mensagem *bestgnsspos*

A implementação da mensagem *bestgnsspos* foi realizada tendo como base a mensagem *bestpos*, estas mensagens têm a mesma estrutura.

Principais etapas da implementação da mensagem *bestgnsspos* na comunicação com o sensor:

- Criação do ficheiro *NovatelGNSSPosition.msg*

```
# Parsed Best GNSS Position or Omnistar XP or HP pos data from Novatel OEM6 receiver
Header header

NovatelMessageHeader novatel_msg_header

string solution_status
string position_type

# Position Data
float64 lat
float64 lon
float64 height

float32 undulation
string datum_id

# Accuracy Statistics (units?)
float32 lat_sigma
float32 lon_sigma
float32 height_sigma
string base_station_id
float32 diff_age
float32 solution_age
uint8 num_satellites_tracked
uint8 num_satellites_used_in_solution
uint8 num_gps_and_glonass_l1_used_in_solution
uint8 num_gps_and_glonass_l1_and_l2_used_in_solution

NovatelExtendedSolutionStatus extended_solution_status

NovatelSignalMask signal_mask
```

Listagem C.1: Ficheiro *NovatelGNSSPosition.msg*

- Criação da ordem de análise da informação recebida no ficheiro *novatel_gps.cpp*.

```

bool NovatelGps::Connect(const std::string& device, ConnectionType connection)
{
    (...)
    NovatelMessageOpts opts;
    opts["bestgnsspos"] = 0.05;
    (...)
}

NovatelGps::ReadResult NovatelGps::ParseNovatelSentence(const NovatelSentence& sentence,
    const ros::Time& stamp) throw(ParseException)
{
    (...)
    else if (sentence.id == "BESTGNSSPOSA")
    {
        novatel_gps_msgs::NovatelGNSSPositionPtr positiongnss = bestgnsspos_parser_.ParseAscii(
            sentence);
        positiongnss->header.stamp = stamp;
        novatelnss_positions_.push_back(positiongnss);
        positiongnss_sync_buffer_.push_back(positiongnss);
    }
    (...)
}

```

Listagem C.2: Excerto de código do ficheiro *novatel_gps.cpp*

- Criação do ficheiro *bestgnsspos.cpp* com funções da análise da informação recebida.

```

namespace novatel_gps_driver
{
    (...)
    novatel_gps_msgs::NovatelGNSSPositionPtr
    BestgnssposParser::ParseAscii(const NovatelSentence& sentence) throw(
        ParseException)
    {
        novatel_gps_msgs::NovatelGNSSPositionPtr msg = boost::make_shared<
            novatel_gps_msgs::NovatelGNSSPosition>();
        HeaderParser h_parser;
        msg->novatel_msg_header = h_parser.ParseAscii(sentence);

        if (sentence.body.size() != ASCII_LENGTH)
        {
            std::stringstream error;
            error << "Unexpected number of BESTGNSSPOS message fields: " << sentence.body
                .size();
            throw ParseException(error.str());
        }

        bool valid = true;

        msg->solution_status = sentence.body[0];
        msg->position_type = sentence.body[1];
        valid = valid && ParseDouble(sentence.body[2], msg->lat);
        valid = valid && ParseDouble(sentence.body[3], msg->lon);
        valid = valid && ParseDouble(sentence.body[4], msg->height);
        valid = valid && ParseFloat(sentence.body[5], msg->undulation);
        msg->datum_id = sentence.body[6];
        valid = valid && ParseFloat(sentence.body[7], msg->lat_sigma);
        valid = valid && ParseFloat(sentence.body[8], msg->lon_sigma);
        valid = valid && ParseFloat(sentence.body[9], msg->height_sigma);
        msg->base_station_id = sentence.body[10];
        valid = valid && ParseFloat(sentence.body[11], msg->diff_age);
        valid = valid && ParseFloat(sentence.body[12], msg->solution_age);
        valid = valid && ParseUInt8(sentence.body[13], msg->num_satellites_tracked);
        valid = valid && ParseUInt8(sentence.body[14], msg->
            num_satellites_used_in_solution);
        valid = valid && ParseUInt8(sentence.body[15], msg->
            num_gps_and_glonass_l1_used_in_solution);
    }
}

```

```

    valid = valid && ParseUInt8(sentence.body[16], msg->
        num_gps_and_glonass_l1_and_l2_used_in_solution);

    // skip reserved field
    uint32_t extended_solution_status = 0;
    valid = valid && ParseUInt32(sentence.body[18], extended_solution_status, 16);
    GetExtendedSolutionStatusMessage(extended_solution_status, msg->
        extended_solution_status);

    // skip reserved field
    uint32_t signal_mask = 0;
    valid = valid && ParseUInt32(sentence.body[20], signal_mask, 16);
    GetSignalsUsed(signal_mask, msg->signal_mask);

    if (!valid)
    {
        throw ParseException("Invalid field in BESTGNSSPOS message");
    }

    return msg;
}

```

Listagem C.3: Excerto do ficheiro *bestgnsspos.cpp*

- Criação de um parâmetro relativo ao envio da mensagem e dados relativos ao à frequência de receção da informação.

```

namespace novatel_gps_driver
{
class NovatelGpsNodelet : public nodelet::Nodelet
{
public:
    (...)
    , publish_novatelgnss_positions_(false)

    void onInit()
    {
        (...)
        swri::param(priv, "publish_novatelgnss_positions", publish_novatelgnss_positions_,
            publish_novatelgnss_positions_);
        (...)
        if (publish_novatelgnss_positions_)
        {
            novatelgnss_position_pub_ = swri::advertise<novatel_gps_msgs::NovatelGNSSPosition>(
                node, "bestgnsspos", 100);
        }
        (...)
    }

    void Spin()
    {
        (...)
        NovatelMessageOpts opts;
        opts["bestgnsspos" + format_suffix] = polling_period_; // Bestgnss position
        (...)
    }

private:
    bool publish_novatelgnss_positions_;
    ros::Publisher novatelgnss_position_pub_;

    void CheckDeviceForData()
    {
        (...)
        std::vector<novatel_gps_msgs::NovatelGNSSPositionPtr> positiongnss_msgs;
    }
}

```

```

    (...)
    gps_.GetNovatelGNSSPositions(positiongnss_msgs);
    (...)
    if (publish_novatelgnss_positions_)
    {
    for (const auto& msg : positiongnss_msgs)
    {
        msg->header.stamp += sync_offset;
        msg->header.frame_id = frame_id_;
        novatelgnss_position_pub_.publish(msg);
    }
    }
    (...)
}

```

Listagem C.4: Excerto do ficheiro *novatel_gps_nodelet.cpp*

C.2 *Launch file*

Launch file utilizado na comunicação com o sensor.

```

<?xml version="1.0"?>
<launch>
  <!-- Node novatel para comandos relativos a posicao-->
  <node name="novatel_position" pkg="nodelet" type="nodelet" args="standalone
    novatel_gps_driver/novatel_gps_nodelet"><rosparam>
    connection_type: serial
    device: /dev/ttyUSB0
    publish_novatel_positions: true
    publish_novatelgnss_positions: true
    publish_imu_messages: false
    publish_nmea_messages: true
    publish_default_messages: true
    publish_diagnostics: true
    publish_novatel_velocity: true
    frame_id: /gps
  </rosparam>
</node>
  <!-- Node novatel para comandos da unidade inercial -->
  <node name="novatel_imu" pkg="nodelet" type="nodelet" args="standalone
    novatel_gps_driver/novatel_gps_nodelet"><rosparam>
    connection_type: serial
    device: /dev/ttyUSB1
    publish_novatel_positions: false
    publish_novatelgnss_positions: false
    publish_imu_messages: true
    publish_nmea_messages: false
    publish_default_messages: false
    publish_diagnostics: true
    frame_id: /gps
  </rosparam></node>
</launch>

```

Listagem C.5: Ficheiro *drivers_novatel.launch*

Apêndice D

Planeamento

Função utilizada no cálculo da distância entre o veículo e um segmento de reta.

```
double global_path::FindDistanceToSegment(double x1, double y1, double x2, double y2, double
    pointX, double pointY)
{
    double diffX = x2 - x1;
    float diffY = y2 - y1;
    if ((diffX == 0) && (diffY == 0))
    {
        diffX = pointX - x1;
        diffY = pointY - y1;
        return sqrt(diffX * diffX + diffY * diffY);
    }

    float t = ((pointX - x1) * diffX + (pointY - y1) * diffY) / (diffX * diffX + diffY * diffY);

    if (t < 0)
    {
        // point is nearest to the first point i.e x1 and y1
        diffX = pointX - x1;
        diffY = pointY - y1;
    }
    else if (t > 1)
    {
        // point is nearest to the end point i.e x2 and y2
        diffX = pointX - x2;
        diffY = pointY - y2;
    }
    else
    {
        // if perpendicular line intersect the line segment.
        diffX = pointX - (x1 + t * diffX);
        diffY = pointY - (y1 + t * diffY);
    }

    // returning shortest distance
    return sqrt(diffX * diffX + diffY * diffY);
}
```


Apêndice E

Mapviz

E.1 *Plugins* para representação da posição

Nesta secção é apresentada a estrutura de ficheiros que constituem a criação e implementação de um *plugin*.

As principais etapas de implementação de um *plugin* na aplicação *Mapviz* são:

- Criação da interface gráfica.

```
<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
<class>bestpos_config</class>
<widget class="QWidget" name="bestpos_config">
  <property name="geometry">
    <rect>
      <x>0</x>
      <y>0</y>
      <width>400</width>
      <height>222</height>
    </rect>
  </property>
```

Listagem E.1: Excerto do ficheiro *bestpos_config.ui*

- Criação de uma nova *class*.

```
class BestposPlugin : public mapviz_plugins::PointDrawingPlugin
{
  Q_OBJECT

public:
  (...)

  void Draw(double x, double y, double scale);
  void LoadConfig(const YAML::Node& node, const std::string& path);
  void SaveConfig(YAML::Emitter& emitter, const std::string& path);

  QWidget* GetConfigWidget(QWidget* parent);

protected:
  void PrintError(const std::string& message);
  void PrintInfo(const std::string& message);
  void PrintWarning(const std::string& message);

protected Q_SLOTS:
  void SelectTopic();
  void TopicEdited();
```

```

private:
    Ui::bestpos_config ui_;
    QWidget* config_widget_;

    std::string topic_;
    ros::NodeHandle nh;
    ros::Subscriber bestpos_sub_;
    ros::Subscriber inspva_sub_;
    bool has_message_;
    StampedPoint stamped_point;
    swri_transform_util::LocalXyWgs84Util local_xy_util_;

    void BestposCallback(const novatel_gps_msgs::NovatelPositionPtr& bestpos);
    void InspvaCallback(const novatel_gps_msgs::InspvaPtr& imu_inspva);
};

```

Listagem E.2: Excerto do ficheiro *bestpos_plugin.h*

- Criação de um ficheiro com funções associadas aos *widgets* criados na etapa anterior.

```

namespace mapviz_plugins
{
    (...)

    void BestposPlugin::SelectTopic()
    {
        ros::master::TopicInfo topic = mapviz::SelectTopicDialog::selectTopic("novatel_gps_msgs/
            NovatelPosition");

        if (!topic.name.empty())
        {
            ui_.topic->setText(QString::fromStdString(topic.name));
            TopicEdited();
        }
    }

    void BestposPlugin::TopicEdited()
    {
        (...)

        topic_ = topic;
        if (!topic.empty())
        {
            bestpos_sub_ = node_.subscribe(topic_, 1, &BestposPlugin::BestposCallback, this);

            inspva_sub_ = nh.subscribe("inspva", 1, &BestposPlugin::InspvaCallback, this);

            ROS_INFO("Subscribing to %s", topic_.c_str());
        }
    }

    void BestposPlugin::BestposCallback(const novatel_gps_msgs::NovatelPositionPtr& bestpos)
    {
        if (!local_xy_util_.Initialized())
        {
            return;
        }
        if (!has_message_)
        {
            initialized_ = true;
            has_message_ = true;
        }

        stamped_point.stamp = bestpos->header.stamp;
    }
}

```

```

    stamped_point.source_frame = local_xy_util_.Frame();
    double x;
    double y;

    local_xy_util_.ToLocalXy(bestpos->lat, bestpos->lon, x, y);
    stamped_point.point = tf::Point(x, y, bestpos->height);
    lap_checked_ = ui_.show_laps->isChecked();
}

void BestposPlugin::InspvaCallback(const novatel_gps_msgs::InspvaPtr& imu_inspva)
{
    stamped_point.orientation = tf::createQuaternionFromYaw((-imu_inspva->azimuth * (M_PI /
        180.0)) + M_PI_2);

    (...)
    cur_point_ = stamped_point;
}
}

```

Listagem E.3: Excerto do ficheiro *bestpos_plugin.cpp*

E.2 *Plugins* para representação da rota

O *plugin global_planning* tem uma estrutura semelhante aos *plugins* relativos à representação da posição, tendo apenas como diferença, a subscrição de tópicos de um tipo diferente. No excerto de código seguinte é possível observar o tipo de tópicos subscritos e a sua nomenclatura, em cada uma das três vertentes deste *plugin*. É também apresentada a função *GlobalPlanningPlugin::WaypointsCallback* idêntica às funções *GlobalPlanningPlugin::WaypointsStepsCallback* e *GlobalPlanningPlugin::WaypointsPrevNextCallback*.

```

void GlobalPlanningPlugin::TopicEdited()
{
    std::string topic = ui_.topic->text().trimmed().toStdString();
    if (topic != topic_)
    {
        initialized_ = false;
        points_.clear();
        has_message_ = false;
        PrintWarning("No messages received.");

        waypoints_.shutdown();

        topic_ = topic;

        if (!topic.empty())
        {
            if (topic == "/waypoints_full")
            {
                waypoints_ = node_.subscribe(topic_, 1, &GlobalPlanningPlugin::WaypointsCallback, this)
                    ;
            }
            if (topic == "/waypoints_steps")
            {
                waypoints_ = node_.subscribe(topic_, 1, &GlobalPlanningPlugin::WaypointsStepsCallback,
                    this);
            }
            if (topic == "/waypoints_previous_next_wsg84")
            {
                waypoints_ = node_.subscribe(topic_, 1, &GlobalPlanningPlugin::
                    WaypointsPrevNextCallback, this);
            }
        }
    }
}

```

```

        bestpos_sub_ = node_.subscribe("bestpos", 1, &GlobalPlanningPlugin::bestposCallback, this
        );

        ROS_INFO("Subscribing to %s", topic_.c_str());
    }
}

(...)

void GlobalPlanningPlugin::WaypointsCallback(const std_msgs::Float64MultiArrayConstPtr &array)
{
    if (!local_xy_util_.Initialized())
    {
        return;
    }
    if (!has_message_)
    {
        initialized_ = true;
        has_message_ = true;
    }

    int size = array->data.size();
    int size_ = size / 2;
    int ii = 0;

    double lat_way[size_];
    double lon_way[size_];

    for (int i = 0; i < size_; i++)
    {
        lat_way[i] = array->data[i];
    }
    for (int i = size_; i < size; i++)
    {
        lon_way[ii] = array->data[i];
        ii++;
    }

    for (int i = 0; i < size_; i++)
    {
        StampedPoint stamped_point;

        (...)

        stamped_point.stamp = stamped_point_point.stamp;
        stamped_point.source_frame = local_xy_util_.Frame();

        double x = 0;
        double y = 0;
        local_xy_util_.ToLocalXy(lat_way[i], lon_way[i], x, y);

        stamped_point.point = tf::Point(x, y, 0);

        (...)

        cur_point_ = stamped_point;
        swri_transform_util::LocalXyWgs84Util local_xy_util_;
    }
}

```

Listagem E.4: Excerto do ficheiro *global_planning_plugin.cpp*

E.3 Sistema de eixos do veículo

A secção mostra a criação do nodo `imu_transform_publisher.cpp`, no *package* `swri_transform_util`, responsável pela criação do sistema de eixos central do veículo - *base_link_imu*.

```
using namespace swri;

namespace swri_transform_util
{
class ImuTransformPublisher : public nodelet::Nodelet
{
private:
    ros::Subscriber imu_sub_;
    ros::Subscriber bestpos_;

    tf::TransformBroadcaster tf_imu;

    swri_transform_util::TransformManager tf_manager_imu_;

    std::string veh_frame_id_;
    std::string global_frame_id_;
    tf::Transform transform;

public:
    void onInit();

    void HandleImu(const novatel_gps_msgs::InspvaPtr& imu_inspva);
    void HandleImuxy(const novatel_gps_msgs::NovatelPositionPtr& bestpos);
};

void ImuTransformPublisher::onInit()
{
    ros::NodeHandle prive = getPrivateNodeHandle();

    swri::param(prive, "child_frame_id", veh_frame_id_, std::string("base_link_imu"));
    swri::param(prive, "parent_frame_id", global_frame_id_, std::string("map"));

    imu_sub_ = getNodeHandle().subscribe("inspva", 100, &ImuTransformPublisher::HandleImu, this);

    bestpos_ = getNodeHandle().subscribe("bestpos", 100, &ImuTransformPublisher::HandleImuxy,
        this);
    tf_manager_imu_.Initialize();
}

void ImuTransformPublisher::HandleImu(const novatel_gps_msgs::InspvaPtr& imu_inspva)
{
    double yaw = (90.0 - imu_inspva->azimuth) * swri_math_util::_deg_2_rad;
    yaw = swri_math_util::WrapRadians(yaw, swri_math_util::_pi);

    tf::Quaternion orientation;
    orientation.setRPY(0, 0, yaw);
    transform.setRotation(orientation);
}

void ImuTransformPublisher::HandleImuxy(const novatel_gps_msgs::NovatelPositionPtr& bestpos)
{
    swri_transform_util::Transform to_local_xy;

    if (tf_manager_imu_.GetTransform(global_frame_id_, swri_transform_util::_wgs84_frame, ros::
        Time(0), to_local_xy))
    {
        tf::Vector3 position(bestpos->lon, bestpos->lat, bestpos->height);
        position = to_local_xy * position;

        transform.setOrigin(position);
    }
}
}
```

```
tf_imu.sendTransform(tf::StampedTransform(transform, bestpos->header.stamp,
    global_frame_id_, veh_frame_id_));
}
}
}
```

Listagem E.5: Excerto de código do ficheiro *imu_transform_publisher.cpp*