

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



Sistema de Posicionamento Baseado em Fusão Sensorial de Dados

Daniel Henrique Moreira Pinto Duque Barbosa

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador: Rui Esteves Araújo

Co-orientador: António Lopes

22 de julho de 2016

Resumo

Os sistemas de posicionamento são uma componente importante de diferentes sistemas de engenharia. Em particular, são importantes em sistemas que envolvam o controlo de movimento, os quais dependem da medida ou estimativa precisa de certas grandezas e da caracterização das mesmas, permitindo o posicionamento e a identificação de parâmetros de movimento do sistema. Na literatura, existe um grande leque de aplicações que utilizam diferentes algoritmos e diferentes tipos de cadeias de medição.

Nesta dissertação, apresenta-se um algoritmo de fusão sensorial para estimar o posicionamento de um veículo terrestre, baseado no Extended Kalman Filter (EKF), e com utilização de medições inerciais e de medições obtidas com um recetor Global Positioning System (GPS). Realiza-se uma análise da informação recolhida pelos dispositivos de medição e de outros métodos de fusão sensorial, identificando-se o comportamento e o erro das medições dos dispositivos, o que permitirá uma integração dos dispositivos de medição com o algoritmo de fusão sensorial.

O método proposto é aplicado num caso de estudo, de um veículo terrestre. O movimento deste veículo é caracterizado por um conjunto de equações matemáticas, que relaciona as entradas do sistema com a dinâmica das variáveis de estado. O EKF é desenvolvido com base neste modelo, que descreve o movimento do veículo, obtendo-se uma estimativa das variáveis de estado e um processamento das medições.

Estima-se o estado de um veículo que inclui a velocidade longitudinal, lateral e angular, e as forças laterais exercidas no veículo. Obtém-se, além das estimativas, uma filtragem das medições realizadas, resultando num sinal dos dispositivos de medição com ruído reduzido. Desenvolveu-se um segundo algoritmo que estima um conjunto maior de informação, incluindo a direção e posição do veículo.

Os algoritmos serão submetidos a sinais de teste obtidos com um modelo CarSim de alta fidelidade, comparando-se a informação dos dispositivos de medição do modelo com o resultado da estimação do algoritmo.

Abstract

Positioning systems are an important system in autonomous vehicles, existing a large range of application of different algorithms and different kind of information obtained from sensors to achieve the estimation of the position and other parameters of the vehicle's movement. In this dissertation it is presented a sensor fusion algorithm to estimate positioning information of a land vehicle, using inertial measurements combined with Global Positioning System (GPS) information, based on the Extended Kalman Filter algorithm (EKF). The proposed method will be applied to a case study, tracking and estimating the state vector of a vehicle that includes the longitudinal and lateral velocity, the yaw rate and the lateral forces applied on the vehicle. A second algorithm capable of estimating more information about the vehicle, like the heading and position, will be developed. In order to validate the estimation of the proposed sensor fusion algorithm it will be simulated with information obtained with a high-fidelity CarSim model, comparing the sensor information obtained from this model with the results of the algorithm.

Agradecimentos

Deixo aqui o meu profundo agradecimento a todos os meus colegas, amigos e família que me apoiaram nesta etapa académica. Foi um período longo, que nunca teria sido concluído não fosse o suporte e encorajamento recebido ao longo deste trabalho de dissertação. Dedico o curso de Engenharia Eletrotécnica e de Computadores, que agora termino, a todos os que me acompanharam e ajudaram neste período importante e determinante da minha vida.

Agradeço ao meu orientador, o Professor Rui Esteves Araújo, pela tese de mestrado que me colocou, e ao meu co-orientador, António Lopes, pelo acompanhamento dado ao longo do semestre. Foram fundamentais para a conclusão da dissertação e para a realização de todo o trabalho a ela associada.

Conteúdo

1	Introdução	1
1.1	Enquadramento	1
1.2	Objetivos	2
1.3	Motivação	2
1.4	Estrutura da Tese	3
2	Revisão Bibliográfica - Estado da Arte	5
2.1	Tecnologias de Medição	5
2.1.1	Sistemas de Localização por Satélite	5
2.1.2	Sistemas de Navegação Inerciais	11
2.2	Fusão Sensorial	16
2.2.1	Conceito	16
2.2.2	Algoritmos	17
2.2.3	Aplicações	22
3	Algoritmo de Fusão Sensorial	25
3.1	Modelo do Veículo	25
3.2	Extended Kalman Filter	29
3.2.1	Algoritmo Proposto 1	33
3.2.2	Algoritmo Proposto 2	36
3.2.3	Discussão	39
4	Implementação do Algoritmo	41
4.1	Requisitos do Sistema	41
4.2	Hardware	42
4.2.1	Arquitetura do Hardware	42
4.2.2	Dimensionamento do Hardware	43
4.3	Caracterização dos Dispositivos de Medição	45
4.3.1	Metodologia	45
4.3.2	Erro	47
4.3.3	Correção do Erro	52
4.3.4	Transformadas	54
4.4	Comunicação	57
4.5	Discussão	60
5	Resultados	63
5.1	Resultados de Simulação	63
5.1.1	Double Line Change: DLC	67

5.1.2	Jturn	74
5.1.3	Discussão	77
6	Conclusão e Trabalho Futuro	79
6.1	Conclusão	79
6.2	Trabalho Futuro	80
A	Script para Comunicação Série com Teensy	81
A.1	Código Python	81
B	Erro da Projeção Equirectangular	85
B.1	Código Matlab	85
	Referências	87

Lista de Figuras

1.1	Componentes de um sistema de fusão sensorial	2
2.1	Triangulação da posição com sinais emitidos por satélites [1]. Pseudo-range é uma margem de precisão, necessário devido ao erro das medições	6
2.2	Ambiguidade na triangulação quando não há satélites suficientes	6
2.3	Representação de acontecimento dos erros Multipath e Shadowing [2]	8
2.4	Frequência de funcionamento do GPS e do GLONASS [3]	9
2.5	Comparação de precisão ente o GLONASS e o GPS [4]	10
2.6	Precisão prevista do GLONASS [4]	10
2.7	Exemplo de um giroscópio MEMS de três eixos, usado no iPhone 4S (Chipworks) [5]	12
2.8	Exemplo de um acelerómetro simples [6]	13
2.9	Representação do erro de fator de escala [6]	15
2.10	Representação do erro de cross-coupling [6]. Os eixos do acelerómetro (x^{accel} , y^{accel} , z^{accel}) e os eixos do giroscópio (x^{gyro} , y^{gyro} , z^{gyro}) não estão alinhados com o referencial do IMU (x, y, z)	16
3.1	Referencial de coordenadas do veículo	26
3.2	Forças aplicadas nas rodas frontal esquerda e traseira direita	27
3.3	Execução do algoritmo filtro duplo	32
4.1	Representação simplificada do sistema a implementar	42
4.2	Esquema da estrutura de dados usada na Variância de Allan [7]	46
4.3	Desvio de Allan para os 3 eixos do acelerómetro e do giroscópio	49
4.4	Ampliação do Desvio de Allan para os 3 eixos do acelerómetro e do giroscópio, com adição de uma reta (a magenta) de declive $-0,5$	49
4.5	Medição do percurso realizado pelo par GPS+Módulo (a preto) e de um <i>smartphone</i> (a vermelho)	51
4.6	Gráfico eixo vs eixo das medições do magnetómetro: (a) eixo x vs y , (b) eixo z vs x , (c) eixo y vs z	53
4.7	Mapa com os pontos definidos para calcular o erro da projeção	56
4.8	Exemplo de uma frame da comunicação por I ² C [8]	58
4.9	Exemplo de escrita por parte de um master no protocolo I ² C [8]	58
4.10	Exemplo de leitura por parte de um master no protocolo I ² C [8]	59
4.11	Comunicação multi-slave através do protocolo SPI [9]	59
4.12	Exemplo de comunicação no protocolo SPI [9]	60
4.13	Esquema da montagem do hardware utilizado	61
4.14	Medição de osciloscópio da comunicação por I ² C do master com: (a) IMU; (b) GPS	61
5.1	<i>Steering</i> das manobras: (a) DLC; (b) Jturn	63

5.2	Velocidade Longitudinal da manobra DLC a 90km/h: (a) Gráfico da estimação; (b) Ampliação do gráfico (a)	68
5.3	Velocidade Lateral da manobra DLC a 90km/h: (a) Gráfico da estimação; (b) Ampliação do gráfico (a)	68
5.4	Erro de estimação da Velocidade Lateral da manobra DLC a 90km/h dos dois algoritmos	69
5.5	Velocidade Angular da manobra DLC a 90km/h: (a) Gráfico da estimação; (b) Ampliação do gráfico (a)	69
5.6	Erro de estimação da Velocidade Angular da manobra DLC a 90km/h dos dois algoritmos	70
5.7	Aceleração Lateral da manobra DLC a 90km/h: (a) Gráfico da estimação; (b) Ampliação do gráfico (a)	70
5.8	Força Lateral da manobra DLC a 90km/h: (a) Força Lateral Dianteira; (b) Força Lateral Traseira	71
5.9	Ampliação da estimação da Força Lateral Dianteira da manobra DLC a 90 km/h dos dois algoritmos	71
5.10	Erro da estimação da Força Lateral Dianteira da manobra DLC a 90 km/h dos dois algoritmos	71
5.11	Estimação da Direção do Veículo da manobra DLC a 90km/h	72
5.12	Estimação do Slip Angle do Veículo da manobra DLC a 90km/h	72
5.13	Estimação da posição da manobra DLC a 90km/h: (a) Posição em x; (b) Posição em y; (c) Posição x vs y; (d) Ampliação do gráfico (c)	73
5.14	Erro da Estimação da Velocidade Lateral dos dois algoritmos para a manobra DLC: (a) a 60km/h; (b) a 120km/h; (c) a 160km/h	74
5.15	Estimação da Velocidade Lateral dos dois algoritmos para a manobra Jturn: (a) a 60km/h; (b) a 90km/h; (c) a 120km/h	75
5.16	Erro da Estimação da Velocidade Lateral dos dois algoritmos para a manobra Jturn: (a) a 60km/h; (b) a 90km/h; (c) a 120km/h	76
5.17	Erro da Estimação da Velocidade Angular dos dois algoritmos para a manobra Jturn: (a) a 60km/h; (b) a 90km/h; (c) a 120km/h	77

Lista de Tabelas

2.1	Representação do desvio padrão dos erros do GPS [1]	9
2.2	Classificação dos IMU de acordo com o bias que apresentam [6]	15
4.1	Bias do IMU	48
4.2	ARW do giroscópio e VRW do acelerómetro	50
4.3	Instabilidade de bias do IMU	51
4.4	Bias fixo do IMU	53
4.5	Erro da projeção equiretangular	57
5.1	Ruído das medições	64

Abreviaturas e Símbolos

AGV	Automated Guided Vehicle
ARW	Angle Random Walk
AUV	Automated Aerial Vehicle
DLC	Double Line Change
DM	Direção do Movimento
DMP	Digital Motion Processor
DV	Direção do Veículo
EKF	Extended Kalman Filter
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
Hz	Hertz
I ² C	Inter-Integrated Circuit
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
MAP	Maximum a Posteriori
MEMS	Micro-Electro-Mechanical Systems
ML	Maximum Likelihood
MOEMS	Micro-Opto-Electro-Mechanical Systems
PIGA	Pendulous Integrating Gyroscopic Accelerometer
PSD	Power Spectral Density
RAM	Random-Access Memory
SMC	Sequential Monte Carlo
SPI	Serial Peripheral Interface
SRAM	Static Random-Access Memory
TOA	Time of Arrival
UKF	Unscented Kalman Filter
VRW	Velocity Random Walk

Capítulo 1

Introdução

1.1 Enquadramento

O grau de maturação dos dispositivos de medição tem permitido uma evolução evidente nas técnicas de controlo de sistemas. No entanto, existem aplicações de complexidade elevada, cujo nível de medição atual não possibilita a caracterização satisfatória do sistema. De forma a permitir tratar problemas de mais complexos e com requisitos mais restritos, aplicam-se técnicas de fusão de informação de múltiplos dispositivos de medição com o intuito de melhorar a exatidão das medições do sistema. Estas técnicas são denominadas de fusão sensorial.

Através da fusão sensorial da informação de diferentes dispositivos de medição, obtém-se uma medição cuja a informação seria impossível obter com os dispositivos de forma individual, melhorando significativamente a informação individual medida.

O termo "fusão sensorial" começou a ser usado, num contexto académico e experimental, nos anos 80, tornando-se cada vez mais proeminente com o passar dos anos. A fusão sensorial surgiu da necessidade de integração de informação e de melhoria da mesma, onde a utilização de métodos estatísticos foi, inicialmente, discutida em [10, 11]. Atualmente, este é um tema largamente explorado, cuja implementação foi desenvolvida através de vários métodos e com aplicações em diversas áreas, como robótica [12, 13, 14, 15, 16], desporto [17], vigilância [18, 19], localização e identificação [20, 19], e biologia [21, 22]. A implementação de fusão sensorial pode ser desenvolvida com dois ou com mais dispositivos de medição, existindo métodos mais indicados para cada caso, e áreas diferentes implicam também métodos diferentes devido à natureza dos dados – por exemplo, quando é necessário combinar informação visual proveniente de mais do que uma câmara, que é um conjunto de informação sensorial completamente diferente do utilizado nesta dissertação, com o objetivo de representar sets diferentes de pontos no mesmo referencial de coordenadas, há opções que permitem uma integração dos dados de forma mais simples do que o Filtro de Kalman [23].

1.2 Objetivos

O objetivo central do trabalho é combinar informação proveniente de diferentes dispositivos de medição de forma a melhorar significativamente a informação obtida pelos mesmos, de tal forma que a integração da informação permita medições com maior exatidão.

Para aplicar o algoritmo de fusão sensorial, será estudado o Filtro de Kalman e variantes do mesmo, pois é um algoritmo comum para este tipo de medição, nomeadamente em sistemas de posicionamento. O algoritmo fará a fusão de dois tipos de informação diferentes, obtidos com um dispositivo de medição inercial (IMU) e um recetor GPS. Do IMU serão adquiridos a aceleração gravitacional nos 3 eixos (x, y, z), as medições angulares do giroscópio e a direção do magnetómetro, e ainda a posição, velocidade e direção proveniente do GPS.

De forma a conseguir um melhor desempenho por parte dos dispositivos de medição, realizar-se-á uma caracterização dos mesmos. Esta caracterização permitirá reduzir alguns problemas dos dispositivos, como o bias.

Pretende-se conceber e implementar o algoritmo de fusão sensorial. Antes da implementação no microcontrolador, é necessário estudar o Filtro de Kalman e as suas variantes.

Na Figura 1.1 apresenta-se os componentes de um sistema de fusão sensorial. Este tipo de sistema engloba cadeias de medição, condicionamento de sinal, interface de saída e realização da fusão sensorial, que inclui hardware de processamento e software.

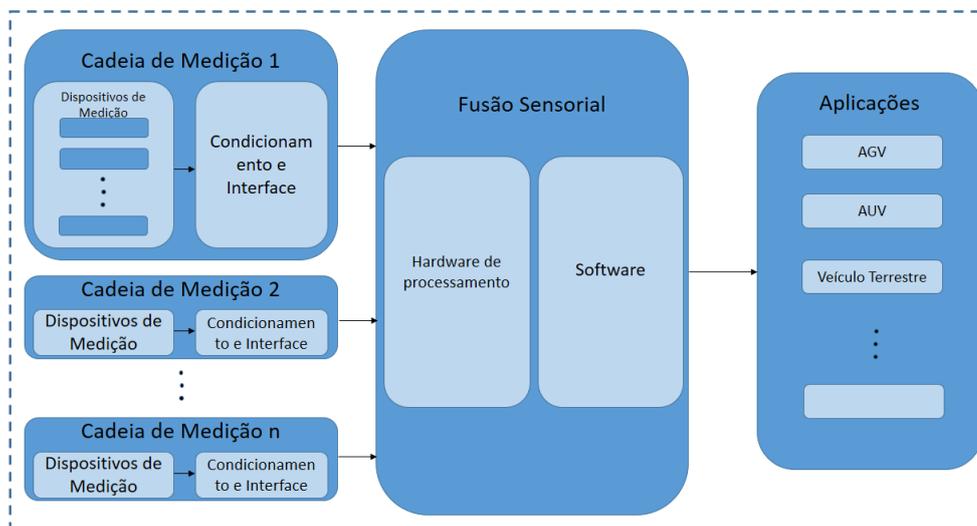


Figura 1.1: Componentes de um sistema de fusão sensorial

1.3 Motivação

De uma forma geral, um sistema móvel autónomo necessita de conhecer a sua posição de forma a deslocar-se corretamente. Para tal, é fundamental um sistema de medida de posição com

boa exatidão.

De forma a obter medições mais precisas e fiáveis ao longo do tempo, é vantajoso o uso de dispositivos de medição de naturezas diferentes, de forma a mitigar as limitações individuais dos mesmos. Se considerarmos um sistema de posicionamento que use dispositivos inerciais apenas, é possível calcular a posição relativa do sistema de forma precisa inicialmente, mas cuja incerteza da medição aumenta ao longo do tempo (drift da medição), sendo impossível eliminar ou corrigir essa incerteza. Por essa razão, é comum emparelhar este tipo de dispositivos de medição com informação proveniente de dispositivos de natureza diferente, como a odometria do sistema (posição do sistema medida através do movimento das rodas) ou com recurso a deteção de balizas através de imagem, de forma a eliminar o erro dos dispositivos inerciais. Num sistema outdoor há, também, a possibilidade de recorrer a sistemas de posicionamento por satélite, como o GPS.

Pretende-se explorar a fusão entre dispositivos de medição inerciais e GPS, que atualmente são dispositivos de medição avançados tecnologicamente, existindo soluções com elevado desempenho a um custo relativamente baixo. Apresentam, no entanto, algumas desvantagens inerentes; Um dispositivo de medição inercial permite obter a posição relativa do sistema, tem uma taxa de aquisição elevada e precisão elevada, mas tem um offset (bias) que provoca um acumulado de erro ao longo do tempo (drift), o que implica uma incerteza elevada. Um recetor GPS permite obter a posição absoluta, a velocidade e a direção do sistema, mas tem uma taxa de aquisição muito inferior à do dispositivo inercial e está dependente da comunicação com os satélites, podendo também levar algum tempo a capturar os satélites necessários para obter a posição. Devido a estas dificuldades técnicas, os dispositivos de medição não podem ser integrados imediatamente no sistema, sendo necessário, primeiro, analisar a informação que estes fornecem.

Tendo em conta as vantagens inerentes aos dispositivos de medição descritos anteriormente, é possível combinar estes, através de algoritmos avançados de fusão sensorial, por forma a obter a posição absoluta do sistema com precisão e com uma taxa de aquisição elevada.

Todos os sistemas que necessitem do conceito de posição podem beneficiar deste processamento de informação sensorial, pois permite obter uma medição atual precisa e até realizar uma previsão ponderada das medições. Variados sistemas em diferentes áreas, como AGVs, UAV e sistemas de augmented reality podem obter um melhor desempenho que o conseguido com a medição de posição mais simples com recurso a um único dispositivo de medição.

1.4 Estrutura da Tese

Nesta dissertação, irá apresentar-se uma revisão bibliográfica, onde se realiza um estudo das tecnologias de medição que serão usadas e dos algoritmos de fusão sensorial. Este estudo é essencial para conhecer as opções que existem e quais as que permitem resultados mais interessantes.

Na secção seguinte, introduz-se o algoritmo de fusão sensorial e o modelo de um caso de estudo, que será o alvo de aplicação do algoritmo.

De seguida, estudar-se-á a implementação do algoritmo desenvolvido. Nesta secção, analisa-se os requisitos dos sistemas, os dispositivos de medição a utilizar e o sistema desenvolvido que realiza a fusão sensorial.

Por fim, apresentar-se-á e discutir-se-á os resultados obtidos com este algoritmo.

Capítulo 2

Revisão Bibliográfica - Estado da Arte

2.1 Tecnologias de Medição

Nesta secção são resumidas várias tecnologias diferentes para a medição de posição, direção, velocidade e aceleração nestes mesmos.

Tratar-se-á dispositivos de medição que fornecem um de dois tipos de localização; localização relativa e localização absoluta. Serão analisadas as vantagens e limitações de cada dispositivo.

2.1.1 Sistemas de Localização por Satélite

Um sistema de navegação por satélite (GNSS) utiliza sinais rádio transmitidos por satélites que se encontram em órbita. Esses sinais são recebidos e interpretados por recetores GNSS, permitindo obter informação de posição, velocidade, tempo e direção [12]. É um sistema de localização absoluta.

O funcionamento destes sistemas baseia-se no conceito de medições TOA (time of arrival), também referido como tempo de propagação, para determinar a posição do utilizador do sistema [2]. Este tipo de medição, aplicada em GNSS, consiste em determinar o tempo que um sinal emitido por um satélite, com posição conhecida, demora a chegar ao recetor. Este tempo é conhecido como o tempo de propagação, a distância do recetor ao emissor é obtida através do tempo de propagação e a velocidade do sinal. No entanto, para se calcular a localização, é necessário comunicar com múltiplos satélites para se obter vários tempos de propagação, podendo-se assim fazer uma triangulação entre os sinais.

Para se obter uma posição sem ambiguidade, será necessário comunicar, no mínimo, com quatro satélites [2]. A necessidade de quatro satélites deve-se à sincronização necessária entre o relógio do recetor e do satélite. O recetor não se encontra síncrono com o relógio do recetor de forma a reduzir a complexidade do mesmo e devido a fenómenos relacionados com a teoria da relatividade. Necessita-se, então, de quatro satélites para obter a latitude, longitude, altura e offset do relógio do recetor em relação ao relógio do sistema.

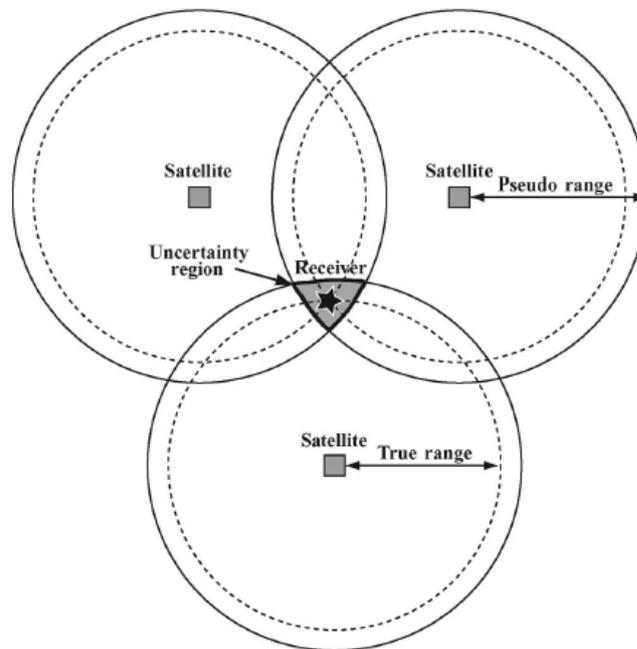


Figura 2.1: Triangulação da posição com sinais emitidos por satélites [1]. Pseudo-range é uma margem de precisão, necessário devido ao erro das medições

A triangulação destas medições permite delimitar uma localização com uma certa precisão e confiança (uma área), do recetor, demonstrado na Figura 2.1. Quanto mais satélites em comunicação, mais precisa será esta localização.

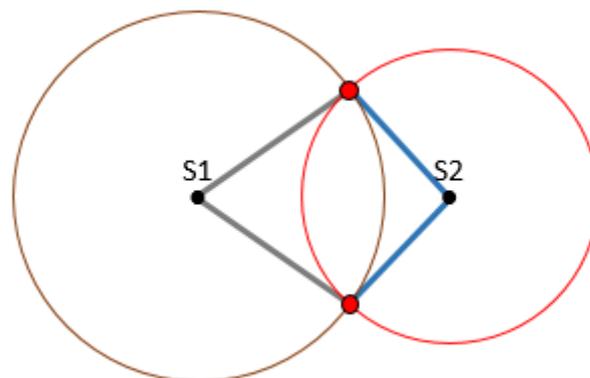


Figura 2.2: Ambiguidade na triangulação quando não há satélites suficientes

Na eventualidade de não existirem pelo menos quatro sinais de satélites ativos, não é possível realizar o cálculo de posição, uma vez que não é possível isolar um único valor, como apresentado na Figura 2.2.

Nestas imagens considera-se que é possível realizar a triangulação com três emissores de sinal, como é normal, mas para a localização por satélite são necessários quatro devido à sincronização de relógios entre o recetor e os satélites.

Há vários GNSS em funcionamento e desenvolvimento. O Global Positioning System (GPS), desenvolvido pelo Departamento de Defesa dos Estados Unidos da América, é o mais preciso e utilizado, seguido do GLONASS, que é o sistema equivalente Russo. Estes dois sistemas terão uma análise mais extensa. O sistema Galileo, desenvolvido por um esforço conjunto da Europa, está em fase de desenvolvimento, tal como o sistema BeiDou, desenvolvido pela China. O GLONASS será apresentado em comparação com o GPS [24].

Este tipo de localização será mais fiável no futuro, quando os sistemas referidos estiverem todos em pleno funcionamento. Atualmente há dois sistemas funcionais e interoperáveis com recurso a certos recetores, disponibilizando 55 satélites no total (31 do GPS, 24 do GLONASS). No futuro, com o pleno funcionamento dos outros dois GNSS, haverá, pelo menos, mais 48 satélites disponíveis, aumentando a precisão e consistência das medições dos recetores.

2.1.1.1 GPS

O GPS é um GNSS desenvolvido pelo Departamento de Defesa dos Estados Unidos da América, cujo projeto começou em 1973 para contornar as limitações de sistemas de localização anteriores. O primeiro satélite foi lançado em 1978 e ficou totalmente operacional em 1995. Inicialmente, foi desenvolvido para uso militar, mas foi disponibilizado para o público geral em 1992 com algumas restrições, que começaram a ser suprimidas a partir de 2000. Atualmente, a sua constelação de satélites é de 31 satélites [25].

O GPS, tal como outros sistemas GNSS, tem limitações intrínsecas à tecnologia usada, que provocam erro nas medições. A maior limitação centra-se na necessária constante comunicação com, no mínimo, quatro satélites para um recetor conseguir calcular a sua posição. Esta limitação impede o uso de um recetor GPS em espaços fechados e torna a temporária perda de satélites, por passar em espaços onde o sinal não chega com a mesma facilidade e a consequente perda de medição de localização, um problema relativamente comum e que é preciso ter em conta ao utilizar os dados provenientes do recetor GPS.

Os erros nas medições podem ser classificados como erros de modo comum e erros de modo não comum [26]. Erros de modo comum são erros consistentes a todos os recetores de GPS numa área grande (50-200 km) e acontecem devido a atrasos na transmissão de sinal pela atmosfera terrestre e, também, devido a erros de relógio de satélite. Erros de modo não comum dependem da localização e especificações técnicas do recetor. Esses erros resultam devido a propagação multipath de sinais rádio e ruído do recetor [1].

Erros atmosféricos são erros de modo comum. A velocidade dos sinais rádios está sujeita ao meio onde as ondas se propagam, sofrendo atrasos em diferentes alturas da nossa atmosfera. Estes atrasos acontecem na Ionosfera, região entre os 70 e 1000 km acima da terra, e na troposfera, que é a região mais baixa da nossa atmosfera (<20 km). Enquanto que o erro provocado pela Ionosfera é fácil de medir e anular, o provocado pela troposfera é mais complicado pois acontece

uma refração de sinal que depende da temperatura, pressão, e humidade. Os modelos usados para calcular o atraso provocado pela troposfera são, por vezes, distantes da realidade, provocando algum erro nas medições [2].

Apesar dos satélites estarem equipados com relógios atómicos muito estáveis para controlar operações realizadas, haverá sempre algum erro que pode atingir os milissegundos [2]. Este é o erro de relógio, que acontece devido a erros relativistas (no espaço o tempo passa mais devagar) e ao movimento dos satélites, que podem deslocar-se ligeiramente da sua órbita planeada.

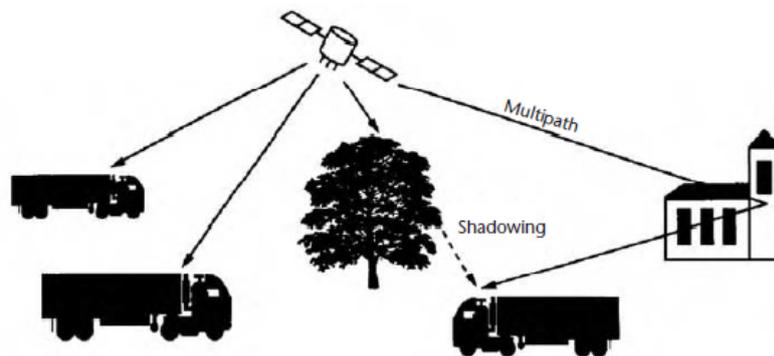


Figura 2.3: Representação de acontecimento dos erros Multipath e Shadowing [2]

Os erros de modo não-comum dependem de recetor para recetor e do espaço onde são utilizados e, com a modernização dos sistemas de GNSS, estes são, por vezes, a maior fonte de erro das medições. Erro de multipath (Figura 2.3) é a receção de reflexões do sinal GPS nos arredores do recetor, recebendo múltiplos sinais que são iguais ao que é recebido diretamente, mas com ligeiros atrasos. Este fenómeno pode distorcer a correlação entre o sinal recebido e a referência gerada no recetor [2], dando medições com erro, que será tanto maior quanto menor for o atraso dos sinais refletidos. O melhor método para evitar este tipo de erros é não utilizar o recetor em locais onde a probabilidade de haver reflexões seja elevada, apesar de haver métodos para reduzir o impacto deste erro. Aliado a este problema da reflexão, há também a possibilidade de acontecer Shadowing (Figura 2.3), que acontece quando o sinal direto do GPS é atenuado por algum objeto não totalmente bloqueador (como uma árvore) e o sinal refletido detetado é mais forte.

A outra fonte de erro de modo não-comum é o ruído, que é introduzido pelo recetor quando realiza as medições. Este ruído acontece, principalmente, devido à antena do recetor que, devido à natureza eletromagnética das ondas rádio, captura radiação eletromagnética produzida por corpos cósmicos e pela atmosfera terrestre que influenciam as medições. Outra fonte de ruído é o ruído térmico, que é um fenómeno físico que acontece com o movimento de eletrões num condutor elétrico [27].

Estes erros são próprios à tecnologia usada, portanto todos os GNSS terão as mesmas limitações. Na tabela 2.1 estão representados os desvios padrão de cada fonte de erro.

Tabela 2.1: Representação do desvio padrão dos erros do GPS [1]

Fonte de Erro		Desvio Padrão (m)
De modo comum	Atmosféricos	7,7
	De relógio	3,6
De modo não-comum	Multipath	0,1-3,0
	Ruído	0,1-0,7

2.1.1.2 GLONASS

O GLONASS é a resposta da Rússia (projeto começado pela União Soviética) ao GPS. O GLONASS lançou o seu primeiro satélite quatro anos após o GPS ter lançado o seu, mas ficou totalmente operacional apenas em 2011, vinte e um anos depois do GPS, devido a vários contratempos, como falta de fundos e avaria de satélites. Usa o mesmo princípio de funcionamento do GPS tendo, portanto, as mesmas limitações.

O GLONASS apareceu da necessidade de uma alternativa ao GPS, para os Russos não estarem dependentes dos Estados Unidos da América para obterem medições de posição. Atualmente, o GLONASS e o GPS têm uma postura de cooperação mútua no desenvolvimento e manutenção de ambos os sistemas. A partir de 2007, o GLONASS disponibilizou acesso total livre a todos os utilizadores internacionais, oferecendo nível de precisão igual à da aplicação militar. O GLONASS só atingiu 100% de cobertura do território Russo em 2010 e, em 2011, com uma constelação de 24 satélites, passou a ter uma cobertura terrestre global. Tem neste momento 28 em órbita, sendo que 24 estão operacionais (com dois deles em manutenção ou em fase de início de operação) [28].

Devido às semelhanças entre ambos, é possível desenvolver recetores que utilizem a informação proveniente dos dois sistemas, aumentando consideravelmente o número de satélites disponíveis para obter medições.

Apesar das semelhanças, o GLONASS tem algumas diferenças técnicas e de performance em relação ao GPS.

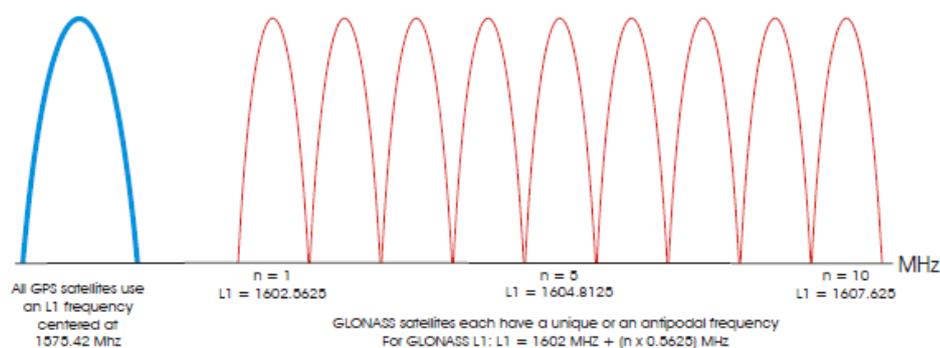


Figura 2.4: Frequência de funcionamento do GPS e do GLONASS [3]

O GLONASS e o GPS têm o mesmo número de satélites em funcionamento, 24, mas usa apenas três planos orbitais enquanto que o GPS usa seis, e tem uma inclinação em relação à linha do Equador superior ($64,8^\circ$ contra 55°). O GLONASS utiliza uma altitude inferior e tem um período de revolução em torno da terra mais rápido (1h55m40s contra 1h58m00s). Estas pequenas variações traduzem-se em performances muito semelhantes, não tendo impacto relevante no desempenho do sistema. Por fim, de forma a não haver interferência de sinais entre os sistemas, estes operam a frequências diferentes. O GPS usa uma frequência centrada nos 1575,42MHz, enquanto que o GLONASS usa várias frequências ligeiramente superiores às do GPS, demonstradas na Figura 2.4 [29, 3].

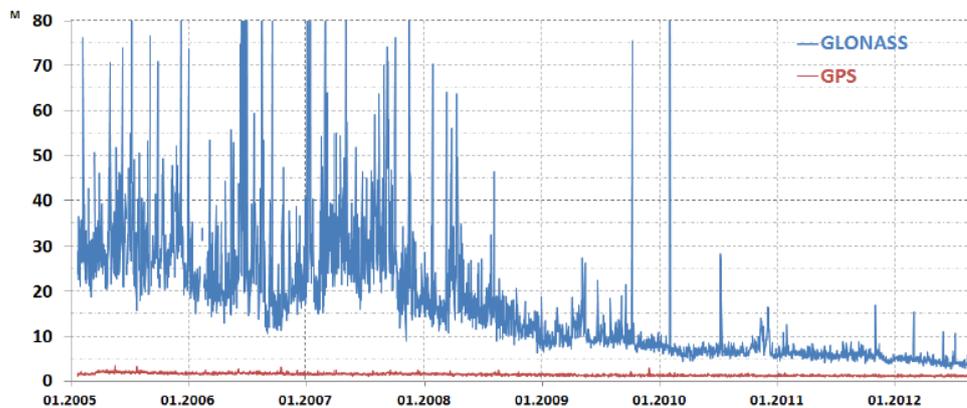


Figura 2.5: Comparação de precisão entre o GLONASS e o GPS [4]

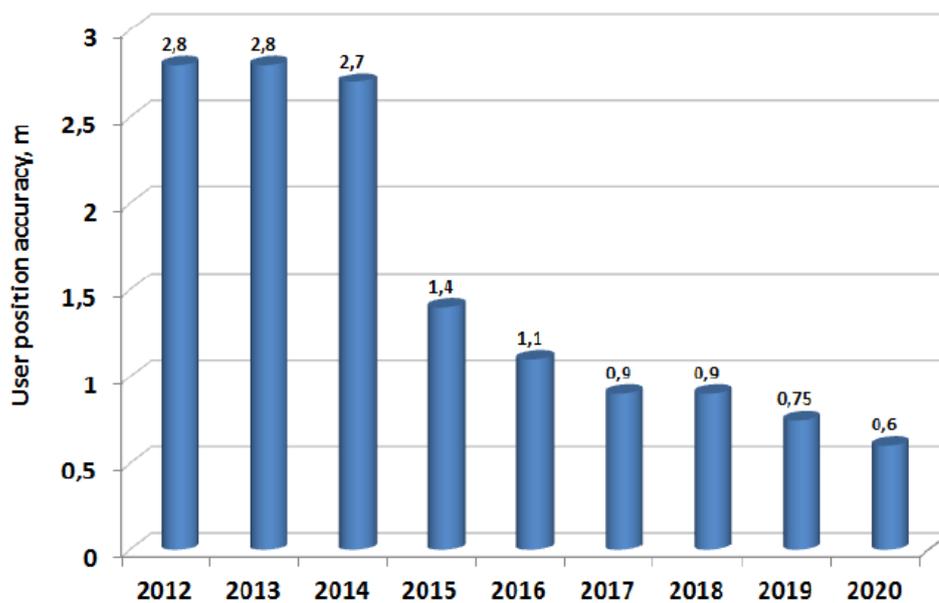


Figura 2.6: Precisão prevista do GLONASS [4]

GLONASS tem, atualmente, uma precisão inferior à do GPS (Figura 2.5), com previsões de conseguir um melhor desempenho (Figura 2.6). Essa melhoria é conseguida com recurso a lançamentos de satélites mais sofisticados para substituir os que estão em fim de vida.

2.1.1.3 Galileo e BeiDou

O Galileo é o GNSS desenvolvido pela Europa e está em desenvolvimento, não estando funcional de momento, tendo apenas 10 satélites operacionais e 12 em órbita. Os serviços iniciais estão previstos para 2016 e o funcionamento na sua máxima capacidade está previsto para 2019 com 24 satélites operacionais, ficando o sistema completo em 2020 com 30 em órbita (6 satélites de reserva).

O BeiDou é o GNSS desenvolvido pela China, concebido em 1980, e está em desenvolvimento, encontrando-se atualmente operacional. Cobre uma pequena área terrestre, incluindo o próprio país e a respetiva região. Tem 18 satélites em órbita, dos quais 10 estão a ser usados para realizar as medições regionais. Começou os testes com um reduzido número de satélites em 2000, começou a cobrir a sua região em 2012 e espera-se que, por 2020, o sistema funcione a uma escala global.

Estes sistemas serão interoperáveis com o GPS e GLONASS, acrescentando um ainda maior número de satélites disponível ao utilizador [30, 31].

2.1.2 Sistemas de Navegação Inerciais

Um sistema de navegação inercial (INS – Inertial Navigation System) combina um conjunto de unidades de medição inercial (IMU – Inertial Measurement Unit), para medir grandezas físicas nos três eixos. Este tipo de sistemas são sistemas de localização relativa.

Um IMU é um dispositivo de medição inercial que produz uma solução independente de navegação a três dimensões, funcionando como dispositivo de um INS. Um IMU é, geralmente, uma combinação de acelerómetros e/ou giroscópios, e por vezes de magnetómetros.

Um dispositivo inercial (acelerómetro, giroscópio ou magnetómetro) apenas produz medições num eixo, sendo necessários três, orientados perpendicularmente, para obter medições nos três eixos da mesma grandeza física [6]. Estes dispositivos realizam medidas sem a necessidade de uma referência externa.

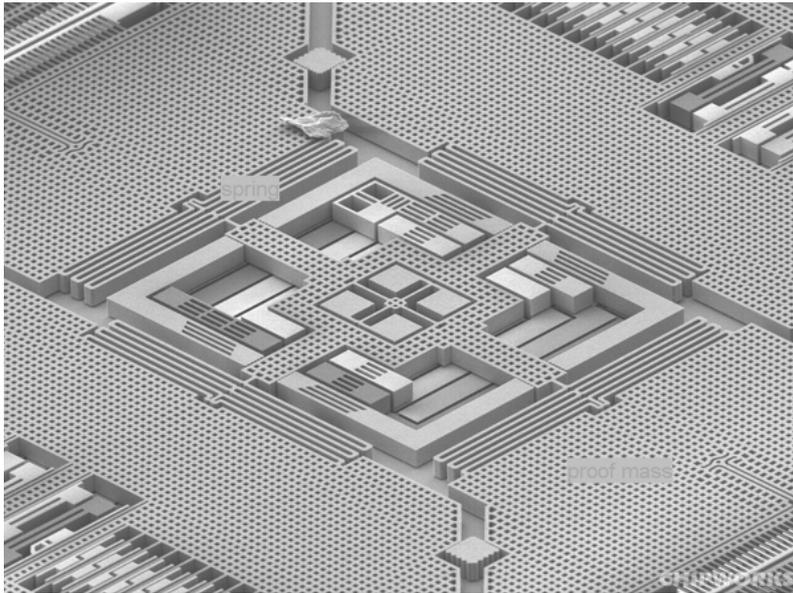


Figura 2.7: Exemplo de um giroscópio MEMS de três eixos, usado no iPhone 4S (Chipworks) [5]

Nos últimos anos verificou-se uma maior utilização [32] deste tipo de IMU devido à tecnologia MEMS (micro-electro-mechanical systems), que é uma tecnologia que permite a construção microscópica de aparelhos (Figura 2.7). Estes dispositivos de medição são mais baratos de produzir mas têm uma performance pior. Essa perda de performance verifica-se numa perda de sensibilidade (alteração mínima detetada numa medição) e num aumento de erro nas medições.

Os MOEMS (Micro-Opto-Electro-Mechanical Systems) também são comuns e estão intrinsecamente ligada ao MEMS, sendo que estes geram as suas medições recorrendo a meios óticos. Esta tecnologia pode, potencialmente, oferecer uma maior performance comparada com os MEMS, no entanto, ainda não atingiram o grau de maturidade necessária [6].

A construção mecânica convencional é ainda muito comum, pois é possível desenvolver dispositivos de medição com uma performance extremamente elevada e com um erro reduzido. No entanto, são mais dispendiosos [6].

Há ainda a tecnologia PIGA (Pendulous Integrating Gyroscopic Accelerometer), que consiste num pêndulo giroscópio montado de forma a que o seu eixo livre esteja fixado num suporte. Esta tecnologia permite grande sensibilidade e precisão numa grande gama de acelerações, sendo muito utilizada em sistemas com dinâmica elevada, como mísseis. No entanto, os MEMS, devido ao baixo custo, apresentam uma solução para sistemas com dinâmicas mais lentas.

2.1.2.1 Acelerómetro

Um acelerómetro usa a medição da força aplicada em cada eixo por forma a inferir a aceleração do sistema. É possível obter a velocidade integrando a aceleração, e a posição integrando a velocidade.

Um dos métodos de medição mais usados por acelerômetros sintetiza-se num sistema com uma massa fixa, num eixo, através de molas. Essa massa, não estando restringida pelas molas no seu movimento, mexe-se ao longo do eixo de acordo com as forças aplicadas no acelerómetro, comprimindo e esticando as molas que a segura. O acelerómetro obtém a aceleração medindo o deslocamento da massa em relação à posição de equilíbrio (acelerómetro em repouso, sem forças aplicadas), que será proporcional à força aplicada.

Este sistema não mede a força gravitacional, uma vez que essa força atua igualmente sobre todos os componentes do acelerómetro (massa e molas), incluindo o próprio acelerómetro, não havendo deslocamento da massa relativo às molas/acelerómetro. Todos os acelerómetros, portanto, medem uma força específica, a aceleração não gravítica [6]. Quando o acelerómetro está alinhado com o eixo da força gravítica, é necessário cancelar a medição de 1g através de calibração.

O acelerómetro não consegue medir alterações de aceleração quando a força aplicada no sistema é inferior à força necessária para fazer deslocar as molas, que são forças tão fracas que não têm influência num sistema autónomo nem precisam de ser detetadas.

Na Figura 2.8 está uma representação simplificada de um acelerómetro, que utiliza o método descrito. O sistema está incompleto, pois seria necessário um suporte perpendicular ao eixo das molas e, também, um amortecimento para limitar a oscilação da massa. Há outros métodos de construção de acelerómetros, como a utilização de um pêndulo ou uma barra vibrante [6], sendo que o princípio de medição através do deslocamento de um elemento devido às forças aplicadas no acelerómetro mantém-se.

2.1.2.2 Giroscópio

Um giroscópio mede a rotação angular. Ao contrário do acelerómetro, cujos métodos de medição são diferentes aplicações do mesmo princípio comum, o giroscópio aplica métodos diferentes igualmente válidos, com diferentes limitações e vantagens. No entanto, a medição realizada por giroscópios com tecnologia MEMS baseia-se, principalmente, num método, o chamado giroscópio vibratório ou giroscópio de estrutura vibratória [6, 33]. Este tipo de giroscópio é constituído por um elemento que é montado de forma a poder realizar movimentos harmónicos simples,

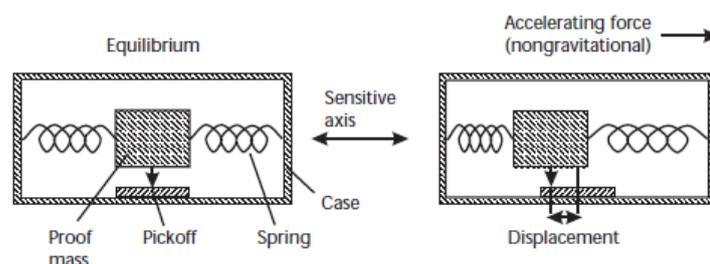


Figura 2.8: Exemplo de um acelerómetro simples [6]

procedendo-se à medição da aceleração de Coriolis do elemento vibratório quando o giroscópio sofre uma rotação. Um exemplo deste tipo de giroscópio está representado na Figura 2.7.

A força de Coriolis é uma força inercial que atua em objetos com um movimento relativo a uma referência rotativa. A aceleração de Coriolis provoca um movimento harmônico no elemento vibratório que se encontra fixo num eixo, sendo que só rotações em torno desse eixo são detetadas e medidas.

As alternativas a este método são os giroscópios de massa rotativa, que operam no princípio da conservação de momento angular, e giroscópios óticos que, a partir da distância percorrida pela luz dentro do giroscópio, medem a rotação angular aplicada no mesmo. Estes giroscópios tiram partido do efeito de Sagnac [6]. Outras tecnologias, como acelerómetros angulares e ressonância magnética nuclear também já foram pesquisadas, não sendo comuns.

2.1.2.3 Magnetómetro

Um magnetómetro mede o campo magnético e pode medir duas vertentes do mesmo: o valor total do campo magnético, ou as componentes vetoriais do campo magnético. De acordo com o sistema onde o magnetómetro está a ser inserido, poderá ser mais interessante uma ou outra medição: por exemplo, se o objetivo for detetar materiais ferromagnéticos num sistema que pode, ele próprio, rodar, poderá ser complicado detetar objetos através das componentes vetoriais do campo magnético, pois irão alterar-se com a rotação do sistema e será difícil separar as duas medições; neste caso, seria muito mais interessante medir o campo magnético total, que permitiria detetar os objetos facilmente [34].

Os magnetómetros vetoriais são afetados por ruído, especificamente ruído $1/f$ ou ruído rosa [35], que é um sinal cuja potência é inversamente proporcional à frequência do sinal. No entanto, tem vindo a ser desenvolvida pesquisa no sentido de reduzir o impacto desse ruído [34].

Dentro dos magnetómetros vetoriais, há várias tecnologias para detetar variações no campo magnético, que permitem diferentes sensibilidades de medição. Magnetómetros do tipo MEMS utilizam o efeito de Hall para fazer as medições, que é uma consequência da força de Lorentz (também conhecido como magnetómetro de efeito de Hall).

O magnetómetro de efeito de Hall é um magnetómetro de baixo custo que tira partido de um fenómeno físico, o efeito de Hall [36], que é a existência de uma diferença de potencial sobre um condutor elétrico, transversal à corrente, e de um campo magnético perpendicular à corrente. A tensão de Hall é criada como resposta à força de Lorentz, que é perpendicular, quer em movimento, quer em direção, aos eletrões que atravessam o campo magnético [34].

2.1.2.4 Erros

Os IMU são invariavelmente afetados por erro, seja qual for a tecnologia dos mesmos e, por norma, os mais dispendiosos apresentam reduções consideráveis desse erro.

Tabela 2.2: Classificação dos IMU de acordo com o bias que apresentam [6]

Classificação IMU	Bias do acelerómetro		Bias do giroscópio	
	mg	ms^{-2}	$^{\circ}hr^{-1}$	$rads^{-1}$
Marítima	0,01	10^{-4}	0,001	5×10^{-9}
Aviação	0,03 – 0,1	$3 \times 10^{-4} - 4 \times 10^{-3}$	0,01	5×10^{-8}
Intermediário	0,1 – 1	$10^{-3} - 10^{-2}$	0,1	5×10^{-7}
Tático	1 – 10	$10^{-2} - 10^{-1}$	1 – 100	$5 \times 10^{-6} - 5 \times 10^{-4}$
Automóvel	> 10	> 10^{-1}	> 100	> 5×10^{-4}

Os erros característicos dos IMU são erros de bias (offset), erros de fator de escala, erros de *cross-coupling* e ruído. A própria construção do IMU pode ser uma origem de erro: os seus componentes (acelerómetro, giroscópio e magnetómetro) não estão colocados exatamente no mesmo local físico, originando medições em referenciais ligeiramente transladados. Esta limitação é, normalmente, corrigida pelo próprio processador do IMU, de forma as medições estarem todas em relação ao referencial do IMU. Os IMU podem ter classificações diferentes de acordo com a grandeza do seu erro de bias, como demonstrado na tabela 2.2.

Erros sistemáticos apresentam quatro componentes: uma contribuição fixa, uma variação dependente da temperatura, uma variação *run-to-run* e uma variação *in-run* [6]. Quer a contribuição fixa, quer a variação dependente da temperatura, podem ser mitigadas ou corrigidas com recurso a calibração. A variação *run-to-run* é um erro que varia cada vez que se usa o dispositivo de medição mas mantém-se constante ao longo da mesma utilização sendo, por isso, corrigível por software. A variação *in-run* altera-se durante a utilização do dispositivo de medição, não sendo possível corrigi-la.

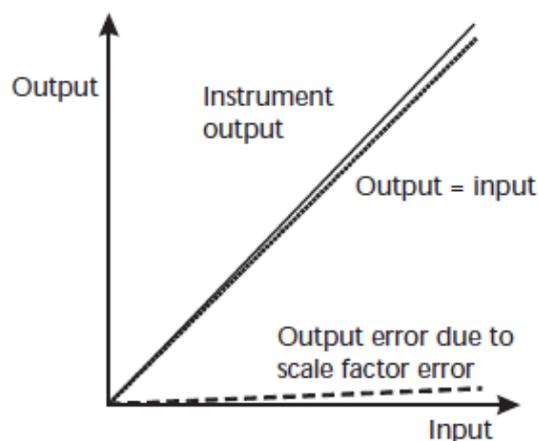


Figura 2.9: Representação do erro de fator de escala [6]

O bias é um erro constante, sendo independente das grandezas físicas que o IMU mede. Pode ser separado em bias estático e bias dinâmico: o bias estático inclui a variação *run-to-run* e o

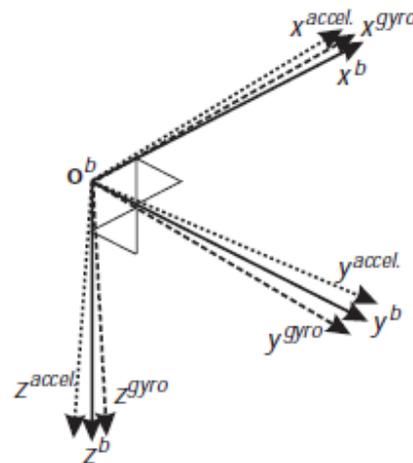


Figura 2.10: Representação do erro de cross-coupling [6]. Os eixos do acelerômetro (x^{accel} , y^{accel} , z^{accel}) e os eixos do giroscópio (x^{gyro} , y^{gyro} , z^{gyro}) não estão alinhados com o referencial do IMU (x, y, z)

bias residual da contribuição física que pode existir após a calibração, variando de utilização para utilização, mas mantendo-se constante ao longo de uma utilização; o dinâmico inclui a variação *in-run* e o bias residual da variação dependente da temperatura que pode existir após calibração.

O erro de fator de escala é o desvio input-output após conversão de unidades, realizada pelo IMU (Figura 2.9). Este erro é proporcional ao valor da grandeza física aplicada no eixo.

O erro cross-coupling é o desalinhamento dos eixos dos componentes do IMU com os eixos ortogonais (Figura 2.10). Isto é uma limitação do processo de construção, onde os componentes não são implementados no IMU com os eixos alinhados exatamente com o referencial do IMU.

2.2 Fusão Sensorial

2.2.1 Conceito

O método de fusão sensorial advém da combinação da informação de múltiplos dispositivos de medição, de forma a obter informação de maior qualidade. Esta mesma informação apresenta características mais interessantes como maior precisão e maior taxa de aquisição, mesmo com o uso de dispositivos comuns. Existem vários algoritmos diferentes de fusão sensorial que procuram explorar fins diferentes, sendo que existem métodos que se adaptam mais facilmente a determinados tipos de aplicações.

Fusão sensorial pode ser usada com o intuito de realizar associação de informação, estimação de estado ou fusão de decisões [37]. Associação de informação confina-se a associar as medições realizadas com os devidos objetos e é utilizada para situações com vários objetos; Estimação de

estado tem como objetivo determinar o estado de um objeto em movimento, através de observações do estado; Fusão de decisão é inferência de alto nível sobre os eventos produzidos por vários dispositivos de medição, ou seja, em vez de se receber informação básica dos dispositivos, recebe-se informação já interpretada e processada em forma de decisão, realizando-se fusão dessa informação de forma a chegar a uma decisão final [38]. Considerando o âmbito desta dissertação, os algoritmos de fusão sensorial a considerar serão apenas métodos de estimação de estado [37].

Fusão de informação é multidisciplinar, abrangendo muitas áreas diferentes, tornando a tarefa de classificação das técnicas difícil. No âmbito desta dissertação, será realizada uma classificação de acordo com a relação entre os dados de input, e do tipo de arquitetura [37].

De acordo com a relação entre os dados de input, pode-se classificar as técnicas de fusão sensorial em três categorias: complementar, redundante e de cooperação. Fusão complementar utiliza informação de naturezas diferentes sobre o mesmo alvo de medição, com o objetivo de obter uma única informação mais global/relevante e de qualidade superior. Fusão redundante utiliza informação do mesmo tipo, com o objetivo de aumentar a confiança nas medições realizadas. Fusão de cooperação combina informação de naturezas diferentes, não com o objetivo de melhorar qualquer uma delas, mas de forma a obter informação mais completa ou com mais elementos diferentes.

A classificação baseada no tipo de arquitetura debruça-se sobre a forma como é realizada, e em que ponto do sistema vai ser realizada a fusão de informação, podendo ser dividida em quatro categorias: arquitetura centralizada, arquitetura descentralizada, arquitetura distribuída e arquitetura hierárquica. Na arquitetura centralizada, o algoritmo de fusão corre num processador central e recebe a informação, sem qualquer tratamento prévio, dos vários dispositivos de medição. Por outro lado, na arquitetura descentralizada há vários nós que realizam a sua própria fusão, não havendo um ponto comum de fusão. Na distribuída, a fusão acontece num processo central, mas recebe a informação de cada dispositivo de medição já processada. Finalmente, na arquitetura hierárquica, a fusão de informação acontece em vários níveis diferentes, onde podem existir arquiteturas descentralizada e distribuída. A arquitetura mais difícil de implementar é a arquitetura descentralizada, pois é mais exigente quer computacionalmente, quer a nível das comunicações. No entanto não há uma arquitetura que se destaque em termos de performance, devendo-se escolher a arquitetura que melhor se adapta aos requerimentos do sistema e à informação disponibilizada pelos dispositivos de medição.

2.2.2 Algoritmos

2.2.2.1 Kalman

O Filtro de Kalman foi originalmente desenvolvido e publicado em 1960 por Rudolph E. Kalman, que se baseou nos trabalhos anteriores de Thorvald Thiele, e num algoritmo semelhante publicado por Peter Swerling em 1958. Em 1961, em conjunto com Richard Bucy, desenvolveu e publicou, novamente, o filtro de Kalman (também conhecido como filtro de Kalman-Bucy), que ainda é utilizado nos dias de hoje. É um algoritmo recursivo que estima o estado de um sistema

dinâmico linear, a partir de medições, e é um filtro muito usado em fusão sensorial [39], devido ao custo computacional relativamente baixo e à facilidade de implementação, acompanhados por uma precisão elevada.

Essencialmente, o filtro de Kalman é um modelo de sistema que, a partir dos sinais de entrada e saída, fornece estimativas de estado que minimizam a covariância do erro da estimativa do estado [40]. O filtro de Kalman está intrinsecamente ligado ao previsor de Kalman, sendo que este estima o estado seguinte, enquanto que o filtro estima o estado no próprio instante. O filtro e o previsor de Kalman aplicam-se quando os ruídos de processo e de medição são ruído branco e de média nula (ruído aleatório).

O algoritmo do filtro de Kalman, apresentado em [40], é o seguinte:

Inicialização:

1. Inicialização:

$$\hat{x}(0) = \mu_x = \mathbf{E}\{x(t)\} \quad (2.1)$$

$$\tilde{P}(0) = \Pi(0) = \mathbf{E}\{[x(0) - \mu][x(0) - \mu]^T\} \quad (2.2)$$

2. Calcular os ganhos do previsor e do filtro de Kalman:

$$K(t) = [A\tilde{P}(t)C^T + S] [C\tilde{P}(t)C^T + R]^{-1} \quad (2.3)$$

$$K_f(t) = \tilde{P}(t)C^T [C\tilde{P}(t)C^T + R]^{-1} \quad (2.4)$$

3. Actualizar estado e matrizes de covariância do erro:

$$\hat{x}(t+1) = A\hat{x}(t) + Bu(t) + K(t) [y(t) - C\hat{x}(t) - Du(t)] \quad (2.5)$$

$$\hat{x}(t|t) = \hat{x}(t) + K_f(t) [y(t) - C\hat{x}(t) - Du(t)] \quad (2.6)$$

$$\tilde{P}(t+1) = A\tilde{P}(t)A^T - K [C\tilde{P}(t)C^T + R]^T K^T + Q \quad (2.7)$$

$$\tilde{P}(t|t) = \tilde{P}(t) - K_f(t)C\tilde{P} \quad (2.8)$$

4. Calcular a previsão $\hat{y}(t)$ e a saída filtrada $\hat{y}(t|t)$:

$$\hat{y}(t) = C\hat{x}(t) + Du(t) \quad (2.9)$$

$$\hat{y}(t|t) = C\hat{x}(t|t) + Du(t) \quad (2.10)$$

As entradas deste algoritmo são:

- $y(t)$ - Saída do sistema;
- $u(t)$ - Entrada do sistema;

- $\hat{x}(0)$ - Estimativa do estado inicial do sistema;
- $\Pi(0)$ - Covariância do erro da estimativa do estado inicial;
- $Q = \mathbf{E} \{q(t)q^T(t)\}$;
- $S = \mathbf{E} \{q(t)r^T(t)\}$;
- $R = \mathbf{E} \{r(t)r^T(t)\}$;
- (A, B, C, D) - Parâmetros do sistema;

Sendo as saídas:

- $\hat{x}(t)$ - Previsão do estado $x(t)$ no instante $t - 1$;
- $\hat{x}(t|t)$ - Estimativa do estado $x(t)$ no instante t ;
- $\hat{y}(t)$ - Previsão da saída $y(t)$ no instante $t - 1$;
- $\hat{y}(t|t)$ - Estimativa da saída $y(t)$ no instante t ;
- $K(t)$ - Ganho do previsor de Kalman;
- $K_f(t)$ - Ganho do filtro de Kalman;
- $\tilde{P}(t)$ - Covariância do erro do previsor de Kalman;
- $\tilde{P}(t|t)$ - Covariância do erro do filtro de Kalman;

O filtro de Kalman é particularmente interessante em sistemas lineares. No entanto, sistemas não lineares são muito comuns e, de forma a obter um desempenho mais interessante nesse tipo de aplicação, foram desenvolvidas variações do algoritmo de Kalman, como o *extended* e o *unscented* [41], que introduzem um passo de linearização e algumas modificações na estimativa de estado. O *extended* utiliza expansões da série de Taylor para linearizar em torno de um ponto de funcionamento, enquanto que o *unscented* calcula um conjunto de pontos sigma, transforma esses pontos para o domínio não-linear do sistema e estima a média e covariância com base nesses pontos.

2.2.2.2 Filtro de Partículas

Filtros de partículas são implementações recursivas do algoritmo Monte Carlo Sequencial (SMC) [39], que são baseados nas regras de Bayes, e apresentam-se como soluções viáveis quando se lida com sistemas não-lineares, ou com ruído não gaussiano.

Este algoritmo utiliza partículas, que são uma distribuição aleatória de amostras, em vez de funções matemáticas para calcular a localização, sendo que cada partícula tem um peso associado, que indica a qualidade dessa partícula. A estimação da posição é, portanto, uma soma pesada de todas as partículas existentes.

Este algoritmo apresenta duas fases, uma onde propaga as partículas (fase de Predição) e outra onde atualiza as mesmas (fase de Atualização). Quanto mais partículas utilizar, mais precisão obtém, o que implica um peso computacional elevado para obter resultados mais precisos [37]. Na fase de predição, as partículas são modificadas de acordo com o modelo existente, tendo em conta uma função de ruído aleatório de forma a simular o efeito do ruído. Na fase de atualização, reavalia-se o peso de cada partícula analisando a informação das medições, removendo partículas com um peso baixo, obtendo-se assim uma estimação mais precisa [37].

O algoritmo genérico seguidamente apresentado está descrito em [37]:

1. Inicialização das partículas:

- (a) Sendo N o número de partículas;
- (b) $X^{(i)}(1) = [x(1), y(1), 0, 0]^T$, para $i = 1, \dots, N$;

2. Fase de predição:

- (a) para cada partícula $i = 1, \dots, N$, estimar o estado $(k+1|k)$ do sistema, usando o estado e o ruído do instante k . Considerando

$$\hat{X}^{(i)}(k+1|k) = F(k)\hat{X}^{(i)}(k) + (\text{cauchy distribution noise})_{(k)}, \quad (2.11)$$

onde $F(k)$ é a matriz de transição do sistema;

3. Calcular o peso das partículas. Para cada partícula $i = 1, \dots, N$:

- (a) calcular a previsão da observação de estado do sistema usando a previsão do estado atual e do ruído no instante k . Considerando

$$\hat{z}^{(i)}(k+1|k) = H(k+1)\hat{X}^{(i)}(k+1|k) + (\text{gaussian measurement noise})_{(k+1)} \quad (2.12)$$

- (b) calcular os pesos de acordo com a distribuição:

$$\text{peso}^{(i)} = N \left(\hat{z}^{(i)}(k+1|k); z^{(i)}(k+1), \text{var} \right) \quad (2.13)$$

- (c) normalizar o valor dos pesos:

$$\tilde{w}^{(i)} = \frac{\text{peso}^{(i)}}{\sum_{j=1}^N \text{peso}^{(j)}} \quad (2.14)$$

4. Fase de Atualização: multiplicar partículas com maior peso e remover as que têm menor peso. O estado atual tem que ser atualizado com o peso das novas partículas:

- (a) calcular a soma dos pesos:

$$W_t^{(i)} = \sum_{j=1}^i \hat{w}^{(j)} \quad (2.15)$$

- (b) gerar variáveis aleatórias distribuídas de forma uniforme $U^{(i)} \sim W(0, 1)$, com o número de passos igual ao número de partículas;
- (c) determinar as partículas que se deve multiplicar e as que se deve remover;

5. Fase de propagação:

- (a) incorporar os novos valores do estado após a Atualização do instante k para calcular o valor do instante $(k + 1)$:

$$\hat{x}^{(i:N)}(k + 1|k + 1) = \hat{x}(k + 1|k) \quad (2.16)$$

- (b) calcular a média posterior:

$$\hat{x}(k + 1) = \text{mean} [x^i(k + 1|k + 1)], i = 1, \dots, N \quad (2.17)$$

- (c) repetir os passos 2 a 5 em cada iteração

Resumindo, filtros de partículas são mais flexíveis do que filtros de Kalman, pois não formulam premissas das densidades de probabilidade a serem aproximadas [39]. Também conseguem lidar com não linearidade e densidades não-gaussianas, quer no processo quer no erro do ruído. No entanto, é necessário um elevado número de partículas para obter uma variância pequena no estimador, tornando este algoritmo bastante pesado computacionalmente e, conseqüentemente, menos atrativo.

2.2.2.3 Maximum Likelihood and Maximum-a-Posteriori

O Maximum Likelihood (ML) é um método de estimação de parâmetros de um modelo estático, baseado em teoria probabilística. Este tipo de estimação (teoria probabilística) é útil quando a variável de estado tem uma distribuição de probabilidade desconhecida [37].

Considerando que a função de verossimilhança é definida como uma função densidade da seqüência de observações z , para dado estado x :

$$\lambda(x) = p(z|x) \quad (2.18)$$

O estimador ML calcula o valor de x que maximiza a função de verossimilhança:

$$\hat{x}(k) = \arg \max_x p(z|x) \quad (2.19)$$

Este método requer o conhecimento do modelo analítico ou empírico do dispositivo de medição, para se conhecer a distribuição prévia e calcular a função de verossimilhança. Também pode provocar um problema de bias, devido à variação da distribuição. No entanto, este problema de bias é mitigado à medida que há um maior número de amostras.

O Maximum-a-Posteriori (MAP) é baseado em teoria Bayesiana. É aplicado quando o parâmetro x a estimar é o output de uma variável aleatória, que tem uma função densidade de probabilidade $p(x)$.

O estimador MAP calcula o valor de x que maximiza a distribuição de probabilidade a posteriori

$$\hat{x}(k) = \underset{x}{\operatorname{arg\,max}} p(x|z) \quad (2.20)$$

Ambos os métodos têm como objetivo estimar o estado x e são equivalentes quando não há informação à priori sobre o mesmo (só há observações). No entanto, o ML assume que x é valor fixo mas desconhecido do espaço dos parâmetros, enquanto que o MAP considera x como um output de uma variável aleatória, com um conhecimento à priori da função de densidade de probabilidade [37].

2.2.3 Aplicações

O filtro de Kalman é tipicamente usado em fusão sensorial, com o objetivo de combinar informação de naturezas diferentes, proveniente de múltiplos dispositivos de medição.

Em [17], é utilizado uma variação do filtro de Kalman (em cascata) para realizar um sistema com um IMU e um recetor GPS de baixo custo, onde foi obtido um resultado superior às alternativas no mercado que usam um sistema baseado em visão. O mesmo tipo de abordagem (Extended Kalman Filter em cascata) foi aplicado com sucesso em sistemas de controlo de UAVs, demonstrado em [14] e de veículos terrestres como em [12], que necessitam de medições precisas de localização. O filtro de Kalman é escalável no sentido em que facilmente se adiciona mais dispositivos de medição, demonstrado em [15], onde se adiciona um magnetómetro a um IMU e GPS, melhorando o desempenho geral do sistema. É um algoritmo utilizado para fazer fusão de informação desde os primórdios do GPS com outros dispositivos de medição [42], demonstrando a tradição e eficácia da sua aplicação neste tipo de desafios.

O filtro de partículas é um filtro mais flexível do que o filtro de Kalman. Em [23], é aplicado um filtro de partículas para detectar e localizar um corpo humano. Em [43], são expostas várias aplicações deste algoritmo, que inclui localização de robôs autónomos e estimação sequencial, entre outros. O filtro de partículas revela-se um algoritmo mais flexível, pois pode ser aplicado numa maior diversidade de problemas, mas tem um custo computacional maior.

O métodos ML e MAP são algoritmos mais simples e, sendo o filtro de Kalman um método probabilístico também, pode-se afirmar que o ML não é capaz de ter a mesma performance do filtro de Kalman. Em [44] foi explorado a utilização do ML como uma alternativa mais simples ao filtro de Kalman. Em [45] este algoritmo foi utilizado para processar e otimizar a informação de múltiplos dispositivos de medição. Em [46], este algoritmo é usado para melhorar a informação e reduzir o erro de um recetor GPS. Este algoritmo não é adequado para realizar fusão sensorial, mas pode ter interesse para tratar a informação dos dispositivos de medição num passo prévio à fusão sensorial.

Estes algoritmos apresentam-se como soluções viáveis para a realização de fusão sensorial, com muita pesquisa acerca da sua utilização já desenvolvida e obtendo resultados interessantes. O filtro de Kalman é o algoritmo de facto standard para este tipo de problema, devido à sua facilidade de implementação e histórico de aplicações. O filtro de partículas é, também, uma solução interessante, mas exige mais poder computacional para obter resultados equiparáveis aos do filtro de Kalman, considerando o número de variáveis necessário para estimar a localização de um sistema. O ML e MAP são algoritmos interessantes não para realizar fusão sensorial, apesar de serem capazes de o realizar, mas sim para processar informação básica proveniente dos dispositivos de medição, num passo anterior à fusão sensorial, realizado por outro algoritmo.

Capítulo 3

Algoritmo de Fusão Sensorial

Nesta dissertação procura-se desenvolver um sistema de posicionamento para aplicação num sistema autónomo, realizando fusão de informação proveniente de diferentes dispositivos de medição de forma a melhorar a informação proveniente dos mesmos. O desafio no panorama tecnológico atual encontra-se no desenvolvimento, não de um sistema preciso apenas, mas sim em conjugar precisão elevada com baixo-custo. Procura-se obter, com recurso a soluções menos dispendiosas (com menor precisão e menor capacidade de processamento), desenvolver um sistema capaz de rivalizar com sistemas e dispositivos de medição singulares mais dispendiosos e com maior precisão.

Para este feito é necessário caracterizar os dispositivos de medição e corrigir o erro característico dos mesmos, realizando também uma configuração e filtragem dependendo da dinâmica do sistema. Será necessário estudar o algoritmo de fusão sensorial e identificar a melhor forma de o implementar, tendo em conta a dinâmica do sistema.

O algoritmo de fusão sensorial será desenvolvido para uma aplicação prática. Considera-se um caso de estudo, onde se pretende estimar informação relacionada com o posicionamento de um veículo terrestre. Neste capítulo será apresentado o modelo do veículo do caso de estudo, seguido de uma secção sobre o algoritmo de fusão sensorial, baseado em Extended Kalman Filter, que aplica o modelo. Na secção seguinte discutir-se-á a implementação do algoritmo, onde se analisa os requisitos de sistema e o hardware a utilizar. Finalmente, serão apresentados os resultados de simulação do algoritmo desenvolvido, terminando-se este capítulo com uma discussão sobre o trabalho desenvolvido e futuro.

3.1 Modelo do Veículo

Nesta secção apresenta-se um modelo matemático de um veículo terrestre de quatro rodas. Este modelo é definido por um conjunto de expressões matemáticas capazes de descrever a dinâmica planar do veículo, relacionando as entradas do sistema, nomeadamente as forças aplicadas e o ângulo de direção, com a dinâmica das variáveis de estado, descrevendo o movimento de um

veículo. A óbvia complexidade e não linearidades do sistema dificulta bastante a tarefa de modelação, pelo que serão realizadas algumas simplificações ao nível do modelo de forma a obter um sistema de equações simples e com um nível razoável de conformidade com a realidade. Na presente análise apenas será estudada a dinâmica nos eixos x , y negligenciando a dinâmica vertical imposta pelo sistema de suspensão do veículo. Esta simplificação permite analisar o veículo como um corpo rígido apresentando movimento de translação apenas nos eixos x e y e rotação no eixo z .

O referencial usado para descrever o sistema do veículo será o referencial do veículo (*body frame*), que está situado no centro de gravidade do veículo (CoG). O eixo x está alinhado longitudinalmente com o corpo do veículo, enquanto o eixo y é perpendicular ao veículo, alinhado lateralmente com o veículo. Na Figura 3.1 representa-se o veículo, o seu referencial e as forças que o modelo considera [47]:

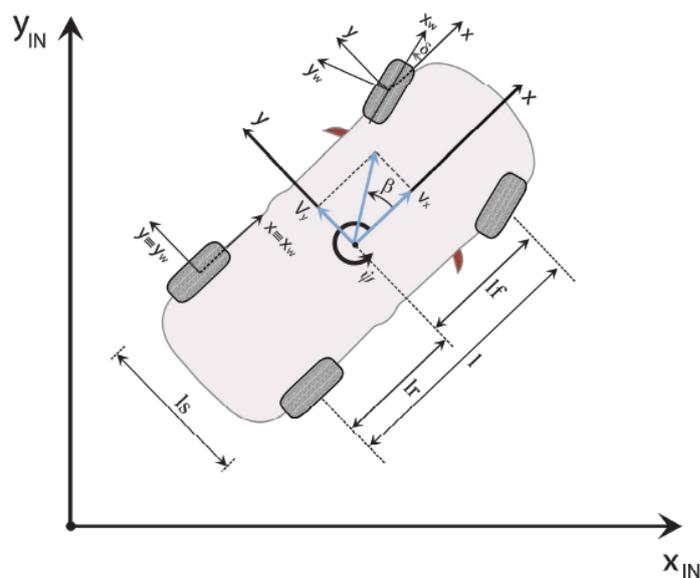


Figura 3.1: Referencial de coordenadas do veículo

onde l_s, l_f, l_r são o comprimento do eixo, a distância entre o eixo dianteiro e o CoG, a distância entre o eixo traseiro e o CoG, β é o *slip angle* e δ é o *steering*. β é o ângulo entre direção física e a direção do movimento do veículo, e o δ é a viragem do veículo. x_{IN} e y_{IN} definem um segundo referencial, que terá o nome "referencial origem", definido no início do movimento do veículo. Um veículo rígido tem dois tipos de movimento, a translação e a rotação. As equações destes movimentos são baseadas nas leis de movimento Newton-Euler, definidas no conjunto de

Equações (3.1) [47]:

$$\dot{v}_x = \frac{1}{m}(F_x) + v_y \dot{\psi} \quad (3.1a)$$

$$\dot{v}_y = \frac{1}{m}(F_y) - v_x \dot{\psi} \quad (3.1b)$$

$$\dot{\psi} = \frac{M_z}{I_z} \quad (3.1c)$$

onde m é a massa total do veículo, I_z o momento inercial, no eixo vertical, do veículo e M_z define o momento total gerado pelas forças aplicadas em cada roda [48]. As variáveis v_x e v_y são a velocidade longitudinal e lateral do veículo, respetivamente, e $\dot{\psi}$ a velocidade angular do veículo. A força F_x representa a soma das forças longitudinais aplicadas no veículo, e F_y a soma de todas as forças laterais que afetam o veículo [48].

Considera-se que o veículo tem tração às quatro rodas, sendo as rodas frontais as que direcionam o veículo. Como o modelo descarta a dinâmica vertical do sistema, pode-se assumir que o comportamento das rodas da esquerda será igual aos da direita, permitindo a análise do veículo como um modelo *single-track*, onde cada eixo é constituído por uma roda apenas [48]. As forças laterais serão definidas para o eixo frontal e traseiro, enquanto que as forças longitudinais são definidas em cada roda devido à dinâmica na rotação.

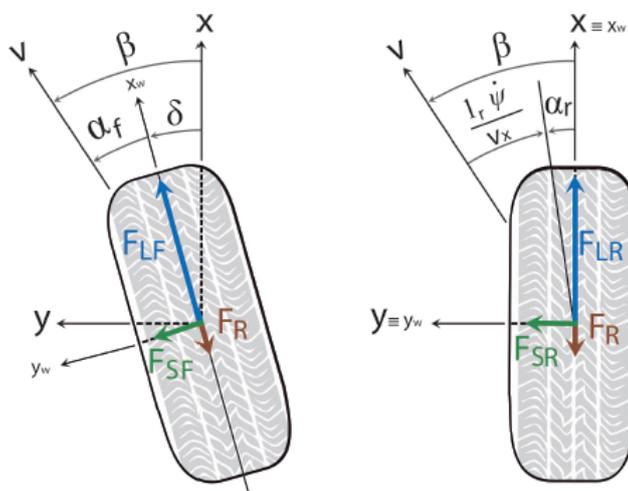


Figura 3.2: Forças aplicadas nas rodas frontal esquerda e traseira direita

Na Figura 3.2 estão representadas as forças aplicadas nas rodas, no referencial da roda, onde F_{Li} e F_{Si} são a força longitudinal e lateral, respetivamente, das rodas, e i é a posição frontal (F) ou traseira (R) da roda [49]. F_r é a força resistiva, que se considera desprezável.

Para se obter o total das forças longitudinais e laterais, expressas na *body frame* da Figura 3.1, pode-se relacionar as forças que atuam em cada roda individualmente, demonstradas na Figura 3.2,

a partir de (3.2):

$$\begin{aligned} F_x &= F_{xF} + F_{xR} = F_{xFL} + F_{xFR} + F_{xRL} + F_{xRR} \\ F_y &= F_{yF} + F_{yR} = F_{yFL} + F_{yFR} + F_{yRL} + F_{yRR} \end{aligned} \quad (3.2)$$

onde F_{xjw} e F_{yjw} são a força longitudinal e lateral que actuam na roda jw – j define a posição dianteira (F) ou traseira (R) da roda, w a posição esquerda (L) ou direita (R) da roda.

As forças longitudinais e laterais que atuam em cada roda são calculadas a partir das Equações (3.3):

$$\begin{aligned} F_{xFL} &= F_{LFL} \cos(\delta) - F_{SFL} \sin(\delta) \\ F_{xFR} &= F_{LFR} \cos(\delta) - F_{SFR} \sin(\delta) \\ F_{yFR} &= F_{SFL} \cos(\delta) + F_{LFL} \sin(\delta) \\ F_{yFL} &= F_{SFR} \cos(\delta) + F_{LFR} \sin(\delta) \\ F_{xRL} &= F_{LRL}, F_{yRL} = F_{SRL} \\ F_{xRR} &= F_{LRR}, F_{yRR} = F_{SRR} \end{aligned} \quad (3.3)$$

onde F_{Ljw} e F_{Sjw} são a força longitudinal e lateral que atuam na roda jw [49].

Substituindo-se (3.3) em (3.2), obtém-se as expressões finais de F_x e F_y , demonstradas em (3.4):

$$\begin{aligned} F_x &= \left((F_{LFL} + F_{LFR}) \cos(\delta) + (F_{LRL} + F_{LRR}) - (F_{SFL} + F_{SFR}) \sin(\delta) \right) \\ F_y &= \left((F_{LFL} + F_{LFR}) \sin(\delta) + (F_{SRL} + F_{SRR}) + (F_{SFL} + F_{SFR}) \cos(\delta) \right) \end{aligned} \quad (3.4)$$

Na formulação que se apresenta, considera-se que a diferença entre a força lateral que atua no eixo dianteiro (F_{SF}) e força lateral traseira (F_{SR}) é desprezável. Assume-se, também, que a força aplicada em cada eixo é a soma das forças aplicadas nas rodas que o constituem, definidas na Equação (3.5a), e que as forças que atuam em cada eixo são iguais, definidas na Equação (3.5b):

$$F_{SF} = F_{SFL} + F_{SFR}, \quad F_{SR} = F_{SRL} + F_{SRR} \quad (3.5a)$$

$$F_{SF} = F_{SR} \quad (3.5b)$$

A força lateral aplicada nas rodas resultam da interação pneu-pavimento, que apresenta uma característica não linear e tem uma forte dependência do coeficiente de atrito do pavimento em questão e dos próprios parâmetros construtivos do pneu [48]. A literatura descreve vários modelos que permitem modelizar este elemento, no entanto, estes recorrem usualmente a modelos não lineares complexos e informação empírica do veículo [50, 51]. A interação pneu-pavimento apresenta uma zona linear que permite uma modelação simples desta força, no entanto, esta mesma simplificação apenas define a força lateral para acelerações laterais baixas [48].

Considerando que M_z é definida na Equação (3.6) [48]:

$$\begin{aligned} M_z &= \left((F_{SFL} - F_{SFR}) \sin(\delta) + F_{LRR} - F_{LRL} + (F_{LFR} - F_{LFL}) \cos(\delta) \right) \frac{l_s}{2} - (F_{SRR} + F_{SRL}) l_r \\ &+ \left((F_{SFR} + F_{SFL}) \cos(\delta) - (F_{LFR} + F_{LFL}) \sin(\delta) \right) l_f \end{aligned} \quad (3.6)$$

e conhecendo todas as forças que actuam no veículo, é possível desenvolver a Equação (3.1), obtendo-se o modelo não linear (3.7) [47]:

$$\begin{aligned}\dot{v}_x &= \frac{1}{m} \left((F_{LFL} + F_{LFR}) \cos(\delta) + (F_{LRL} + F_{LRR}) - (F_{SF}) \sin(\delta) \right) + v_y \dot{\psi} \\ \dot{v}_y &= \frac{1}{m} \left((F_{LFL} + F_{LFR}) \sin(\delta) + (F_{SR}) + (F_{SF}) \cos(\delta) \right) - v_x \dot{\psi} \\ \ddot{\psi} &= \frac{1}{I_z} \left((F_{LRR} - F_{LRL} + (F_{LFR} - F_{LFL}) \cos(\delta)) \frac{l_s}{2} + ((F_{SF}) \cos(\delta) \right. \\ &\quad \left. - (F_{LFL} + F_{LFR}) \sin(\delta)) l_f - (F_{SR}) l_r \right)\end{aligned}\quad (3.7)$$

Além das forças que atuam em cada roda apresentadas previamente, existe ainda uma componente de arrasto (*drag*) associado ao chassi do veículo, definida pela Equação (3.8):

$$F_{DRAG} = \frac{1}{2} \rho C_D A v^2 \quad (3.8)$$

onde ρ é a densidade atmosférica do ar (em kg/m^3), C_D o fator de arrasto, A a área da projeção frontal do veículo (em m^2) e v a velocidade total do veículo (em m/s) [47]. Esta componente será essencial para obter uma estimacão mais precisa da velocidade longitudinal, como será demonstrado na aplicação do algoritmo.

A informação do veículo é obtida com recurso a dispositivos de medição. Com um IMU, obtém-se acesso à aceleração, velocidade angular e direcção do veículo; o GPS permite obter a velocidade do veículo e a direcção do movimento. O algoritmo de fusão sensorial, que utiliza a informação obtida dos dispositivos de medição, estima parâmetros que não são medidos, como as forças laterais, e melhora a informação disponibilizada pelos mesmos.

3.2 Extended Kalman Filter

No Capítulo 2 apresentaram-se algoritmos possíveis e já usados para realizar fusão sensorial. Um dos mais interessantes para este tipo de aplicação o filtro de Kalman. Devido à natureza não linear do sistema, é necessário considerar variações do filtro de Kalman tradicional. Nesta secção são discutidos o Extended Kalman Filter (EKF), o EKF de 2ª Ordem e o Unscented Kalman Filter, para realização de fusão sensorial. Inicialmente é exposto o EKF, sendo este o filtro que será utilizado na dissertação. De seguida serão apresentadas as outras variações referidas.

O EKF é uma evolução natural do filtro de Kalman, sendo uma solução para sistemas não lineares. É uma variação do filtro tradicional que acrescenta um passo de linearização do modelo não linear, de forma a aplicar o ganho de Kalman para sistemas lineares. Um passo fundamental do filtro de Kalman é a propagação da covariância pela dinâmica do sistema; o EKF realiza uma aproximação da covariância, que é depois propagada pela linearização de 1ª ordem do modelo não linear [52], realizando-se a correcção da aproximação da covariância. Naturalmente, devido ao processo de linearização, a aplicação do filtro em casos não lineares é difícil, onde a falta de

convergência do filtro acontece com muita facilidade [52]. O filtro beneficiará de um bom modelo não linear que seja fiel à realidade, de uma boa afinação, e se o modelo for praticamente linear entre os intervalos de atualização do filtro, ou seja, quanto maior a frequência de execução, melhor a aproximação linear, obtendo-se um comportamento superior [53]. Logicamente, um sistema com uma dinâmica não linear elevada poderá ser muito difícil, senão impossível, de estimar, principalmente a uma baixa frequência de execução.

O EKF prevê o próximo estado com base no atual já estimado, corrigindo essa previsão com recurso a medições dos dispositivos de medição, considerando um certo nível de confiança para cada medição definido previamente. Este filtro tem uma capacidade de aprendizagem, que é variável de acordo com alguns valores definidos à-priori, e é afetada pelas correções realizadas a partir das medições: ao corrigir a estimação, obtém-se uma diferença entre o resultado do modelo e das medições, contribuindo para uma melhor estimação do estado. O objetivo do filtro é minimizar o erro da covariância (matriz P) da estimação de estado, que indica o nível de confiança na estimação (quanto menor, mais certeza). Devido à natureza probabilística deste filtro, este tem um comportamento ideal quando o ruído das medições e de processo são ruído branco.

O modelo não linear pode ser definido, de uma forma geral, pelo conjunto de Equações (3.9):

$$\begin{aligned}\dot{X}(t) &= f(t, X(t), U(t)) + w(t) \\ Y(t) &= h(t, X(t), U(t)) + v(t)\end{aligned}\tag{3.9}$$

onde $X(t)$ é o vetor de estado, $U(t)$ é vetor de controlo, $f(t, X(t), U(t))$ é o conjunto de equações não lineares do estado e $h(t, X(t), U(t))$ é o conjunto de equações não lineares das medições.

Assumindo uma discretização *Forward Euler* baseada na expansão de Taylor do modelo não linear, pode-se definir o modelo discreto com o conjunto de Equações (3.10) [54]:

$$\begin{aligned}X_k &= X_{k-1} + \Delta_t f(X_{k-1}, U_{k-1}) \\ Y_k &= h(X_k, U_{k-1})\end{aligned}\tag{3.10}$$

onde Δ_t é o tempo entre duas iterações do filtro, ou seja, o tempo computacional de *step*. w_k e v_k são o ruído de processo e de medição, respetivamente, e são aproximados por uma distribuição gaussiana $w_k \sim N(0, Q)$ e $v_k \sim N(0, R)$, assumindo-se ruído gaussiano de média zero. O ruído de processo pode ter várias fontes, como ruído no vetor de controlo, variação no comportamento de componentes e dispositivos de medição devido a variações de temperatura, ou mesmo alguma componente real que afete o modelo e não esteja modelizada ou tenha sido descartada devido a simplificações. As matrizes Q e R são a matriz de ruído de processo e a matriz de ruído das medições, respetivamente, e a afinação do filtro realizada nesta dissertação passa por variar estas matrizes.

O filtro tem dois passos distintos: o passo de predição, e passo de atualização.

Na fase de predição, calcula-se o novo estado e a matriz de covariância com base nos valores

do ciclo anterior, através da Equação (3.11):

$$X_{k|k-1} = X_{k-1|k-1} + \Delta_t f(X_{k-1|k-1}, U_{k-1}) \quad (3.11a)$$

$$P_{k|k-1} = F_{k-1} P_{k-1|k-1} F_{k-1}^T + Q \quad (3.11b)$$

onde P_k é a matriz de covariância do erro.

Na fase de atualização, calcula-se o ganho K , que determina a extensão da correção das medições e depende da confiança do filtro na estimação. A confiança na estimação é dada pela matriz de covariância P , que será tanto menor quanto maior a confiança na estimação, dando um maior peso ao modelo do que às medições. Segue-se a correção do cálculo do novo estado com base no ganho calculado, corrigindo-se também a matriz de covariância.

As expressões da fase de atualização estão definidas na Equação (3.12):

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R)^{-1} \quad (3.12a)$$

$$X_{k|k} = X_{k-1} + K_k (Z_k - Y_k) \quad (3.12b)$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \quad (3.12c)$$

As matrizes F e H , utilizadas nas Equações (3.11) e (3.12), são a matriz Jacobiano de $f(X_{k-1}, U_{k-1})$ e $h(X_k, U_{k-1})$, respetivamente definidas na Equação (3.13):

$$F_k = \frac{\partial f(X_{k-1}, U_{k-1})}{\partial X_k}, \quad H_k = \frac{\partial h(X_k, U_{k-1})}{\partial X_k} \quad (3.13)$$

estas matrizes implementam a linearização do modelo.

O EKF tem parâmetros que são configuráveis, permitindo a afinação do filtro. É possível alterar o comportamento do filtro, variando as matrizes X_0 , P_0 , Q e R , sendo este processo determinante para se obter uma boa resposta do mesmo. As matrizes fundamentais são as matrizes Q e R , pois são as que definem como o filtro processa as medições e os cálculos do modelo.

Pode-se considerar Q como a matriz que define a contribuição que a correção a partir das medições tem para a estimação de estado, injetando, assim, incerteza no filtro e facilitando a aprendizagem por parte do filtro. É preciso ter em conta que a variação dos valores desta matriz permite obter comportamentos muito diferentes do filtro. Um valor alto de Q (maior ganho) significa que será dado um peso maior às medições, obtendo-se uma estimação mais rápida, mas menos precisa do filtro, passando mais ruído dos dispositivos de medição para a estimação de estado. Se Q tiver valores baixos (menor ganho), será dada uma confiança maior ao cálculo do modelo, dá-se menor peso às medições e obtém-se uma estimação mais precisa, menos ruidosa, mas com um atraso maior. De uma forma geral, a matriz Q permite escolher entre precisão e atraso na estimação, sendo o balanceamento entre as duas um desafio, que depende dos requisitos do sistema e é determinante para obter a convergência.

A matriz R tem como objetivo modelizar o ruído das medições dos dispositivos de medição. Esses valores são o ponto de partida para definir esta matriz, mas poderão ser alterados, uma vez

as matrizes \mathbf{Q} e \mathbf{R} atuam juntas, para obter uma estimaco de maior qualidade.

A matriz \mathbf{P}_0 define o erro da covarincia inicial, e pode ter impacto na velocidade da convergncia inicial. Esta matriz pode ser escolhida de acordo com a confiana que se tem na estimaco da matriz \mathbf{X}_0 , onde uma menor confiana significar num valor \mathbf{P}_0 mais elevado, para inicialmente o filtro convergir mais rapidamente.

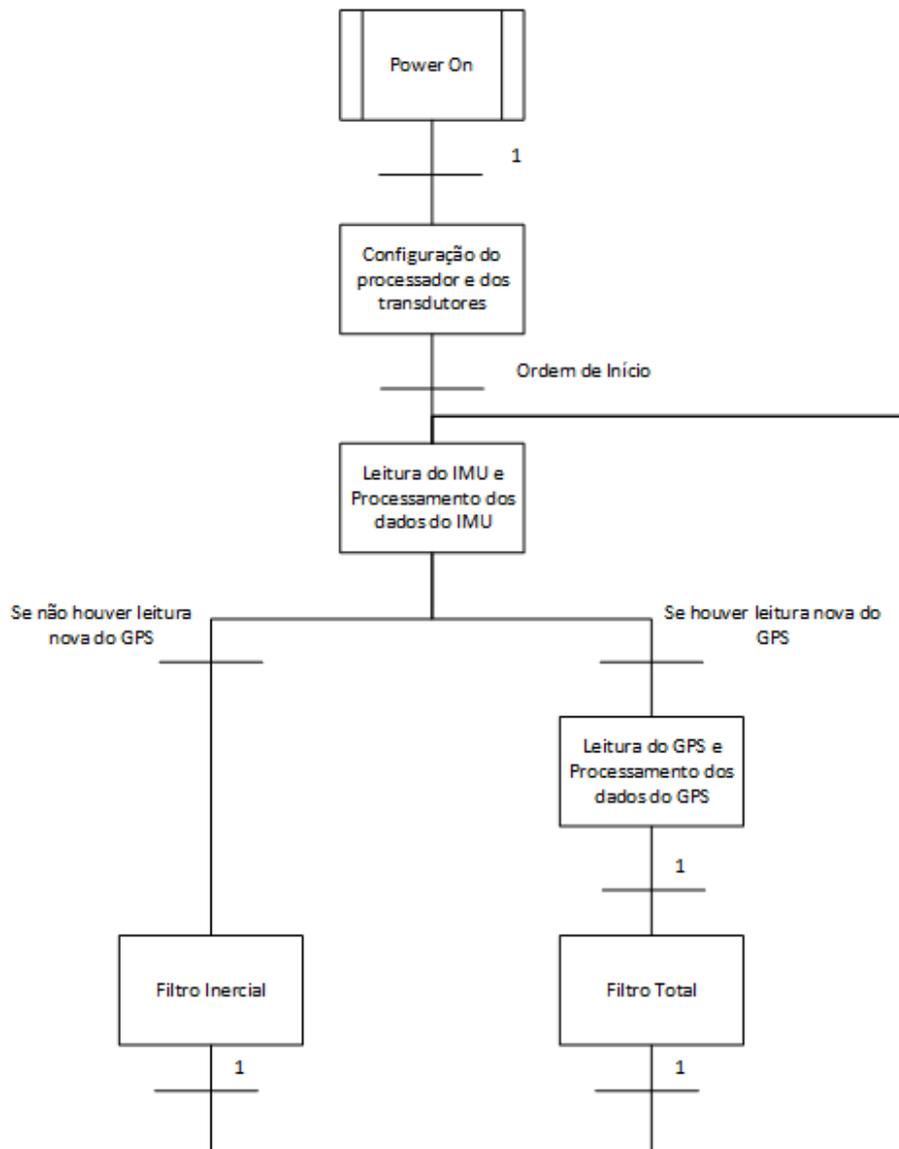


Figura 3.3: Execuo do algoritmo filtro duplo

3.2.1 Algoritmo Proposto 1

A ferramenta Matlab permite o teste do algoritmo em paralelo com o seu desenvolvimento, permitindo também modificar e afinar o algoritmo para diferentes situações e verificar o comportamento do mesmo, percebendo-se a performance do filtro na estimação de estado para manobras mais ou menos exigentes.

Como foi referido na secção anterior, a estimação de estado passa por uma escolha entre o atraso e precisão da estimação, podendo-se favorecer uma estimação mais precisa à custa de um maior atraso, ou uma estimação com menos atraso mas com mais ruído presente e não tão precisa. Encontrar o equilíbrio entre as duas vertentes é importante, pois um atraso elevado pode impedir que o filtro convirja, principalmente em aplicações com uma dinâmica não linear maior, e um ruído elevado pode afetar a estimação das variáveis, resultando também numa não convergência do filtro.

O algoritmo aqui apresentado é caracterizado por um duplo EKF [13]. De forma a conjugar diferentes frequências de aquisição dos dispositivos de medição e conseguir uma frequência de execução superior do algoritmo, desenvolveu-se dois EKF, um que terá acesso às medições dos dispositivos de medição que operam a uma frequência maior (EKF_{IMU}), enquanto que o segundo filtro atuará quando todas as medições estiverem disponíveis (EKF_{TOTAL}). O EKF_{IMU} terá acesso às medições do IMU e o EKF_{TOTAL} terá acesso ao IMU e GPS, atuando no lugar do inercial quando o GPS fizer uma nova leitura. Este processo está representado na Figura 3.3. Ambos os filtros aplicam o mesmo modelo não linear, mas têm algumas diferenças em algumas matrizes e terão configurações diferentes.

Na 1ª iteração do algoritmo, implementou-se o modelo não linear definido na Secção (3.1). Foram definidos os seguintes vetores estado e medição:

$$X_k = \begin{bmatrix} v_x \\ v_y \\ \psi \\ F_{SF} \\ F_{SR} \end{bmatrix}, \quad Y_k = \begin{bmatrix} v_x \\ \dot{v}_y \\ \psi \end{bmatrix}$$

O vetor controlo é definido como:

$$U_k = \begin{bmatrix} F_{LFL} \\ F_{LFR} \\ F_{LRL} \\ F_{LRR} \\ \delta \end{bmatrix}$$

Considerando os vetores X_k e U_k , o modelo pode ser definido pelas Equações (3.14):

$$f(X_k, U_k) = \begin{cases} \frac{1}{m} \left((U_k(1) + U_k(2)) \cos(U_k(5)) + (U_k(3) + U_k(4)) - X_k(4) \sin(U_k(5)) \right) \\ \quad + X_k(3)X_k(2) \\ \frac{1}{m} \left((U_k(1) + U_k(2)) \sin(U_k(5)) + (X_k(5)) + (X_k(4)) \cos(U_k(5)) \right) \\ \quad - X_k(3)X_k(1) \\ \frac{1}{I_z} \left(U_k(4) - U_k(3) + (U_k(2) - U_k(1)) \cos(U_k(5)) \right) \frac{I_s}{2} - (X_k(5)) I_r \\ \quad + \left((X_k(4)) \cos(U_k(5)) - (U_k(1) + U_k(2)) \sin(U_k(5)) \right) I_f \\ 0 \\ 0 \end{cases} \quad (3.14)$$

assumindo-se

$$f_{k_{TOTAL}} f(X_k, U_k) = f_{k_{IMU}} f(X_k, U_k) = f(X_k, U_k) \quad (3.15)$$

O modelo e os vetores estado, medição e controlo são comuns aos dois filtros.

A medição dos dispositivos de medição é definida, para o EKF_{TOTAL}, pelo conjunto de Equações (3.16), e para o EKF_{IMU} pelo conjunto de Equações (3.17).

A diferença entre os modelos das medições é a não presença da medição de velocidade longitudinal no EKF_{IMU}, uma vez que essa medição não é realizada à mesma frequência que as inerciais. A definição de diferentes funções $h(X_k, U_k)$ para os dois filtros obriga ao cálculo de duas matrizes Jacobiano H_k diferentes, enquanto haverá apenas uma matriz F_k .

$$h_{k_{TOTAL}}(X_k, U_k) = \begin{cases} X_k(1) \\ \frac{1}{m} \left((U_k(1) + U_k(2)) \sin(U_k(5)) + X_k(5) + X_k(4) \cos(U_k(5)) \right) \\ X_k(3) \end{cases} \quad (3.16)$$

$$h_{k_{IMU}}(X_k, U_k) = \begin{cases} 0 \\ \frac{1}{m} \left((U_k(1) + U_k(2)) \sin(U_k(5)) + X_k(5) + X_k(4) \cos(U_k(5)) \right) \\ X_k(3) \end{cases} \quad (3.17)$$

A matriz Jacobiano F_k é definida da seguinte forma:

$$F_k = \begin{bmatrix} 0 & X_k(3) & X_k(2) & -\frac{\sin(U_k(5))}{m} & 0 \\ -X_k(3) & 0 & -X_k(1) & \frac{\cos(U_k(5))}{m} & \frac{1}{m} \\ 0 & 0 & 0 & \frac{l_f \cos(U_k(5))}{I_z} & \frac{l_r}{I_z} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

assumindo-se

$$F_{kTOTAL} = F_{kIMU} = F_k$$

As matrizes Jacobiano H_{kTOTAL} e H_{kIMU} são definidas, respetivamente, como:

$$H_{kTOTAL} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\cos(U_k(5))}{m} & \frac{1}{m} \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$H_{kIMU} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\cos(U_k(5))}{m} & \frac{1}{m} \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

As matrizes Jacobiano refletem as diferenças da matriz $h(X_k, U_k)$ para cada filtro.

Neste modelo considera-se que as forças laterais que atuam nos eixos (F_{SF} e F_{SR}) são desconhecidas, como já foi referido na Secção 3.1. Ao serem modelizadas desta forma, o filtro irá estimar estas variáveis com base nas estimações das variáveis de estado, que dependem destas duas forças, e com base na correção dessas estimações a partir de medições. Esta estimacão das forças é conseguida definindo-as como zero, o que significa que não se sabe como estas variáveis variam. Em vez de utilizar um modelo que caracteriza a relação roda-pavimento, utiliza-se um modelo que não precisa de saber essa relação, resultando num algoritmo mais versátil e adaptável a diferentes cenários. Além das forças laterais, estimam-se a velocidade longitudinal, lateral e angular.

Analisando esta implementação, a grande diferença entre os dois filtros prende-se no modelo não linear das medições. No EKF_{TOTAL} dá-se um maior peso à medição da velocidade do que ao modelo, para realizar a correção da estimacão; No EKF_{IMU} , o objetivo é ignorar a medição da velocidade longitudinal e aplicar o modelo sem essa medição.

3.2.2 Algoritmo Proposto 2

Nesta versão, manteve-se a utilização de dois filtros, mas alterou-se ligeiramente o modelo utilizado, acrescentado a estimação de mais variáveis para obter um estado com mais informação, com o objetivo de obter um algoritmo com maior redundância. Foram introduzidas no modelo a estimação das variáveis da direção ψ , *slip angle* β , e posição em x e y . O conjunto de medições terá também, além das quatro medidas referidas na frase anterior, a aceleração longitudinal (a_x) e velocidade lateral (v_y).

Com vista a correção da estimação da velocidade longitudinal, introduziu-se uma componente que, até agora, não fez parte do modelo, a equação do *drag*. O veículo, quando está em movimento, é afetado por uma força, provocada pelo ar, que atrasa o veículo. A introdução desta componente no modelo resulta numa estimação mais precisa do EKF_{IMU} da velocidade longitudinal.

O vetor controlo manteve-se igual. Os novos vetores de estado e medição são, então, definidos como:

$$X_k = \begin{bmatrix} v_x \\ v_y \\ \psi \\ F_{SF} \\ F_{SR} \\ \psi \\ x \\ y \end{bmatrix}, \quad Y_k = \begin{bmatrix} v_x \\ v_y \\ \dot{v}_x \\ \dot{v}_y \\ \psi \\ \psi \\ \beta \\ x \\ y \end{bmatrix}$$

O novo modelo utilizado é definido em (3.18) e reflete as alterações de forma a acomodar o novo estado definido.

A direção ψ é uma integração da velocidade angular. A posição (x, y) , que indica a posição da *body frame*, está expressa em relação ao referencial origem da Figura 3.1. Uma vez que a *body frame* tem a direção do veículo e a posição é calculada a partir de uma integração da velocidade longitudinal e lateral, é necessário aplicar uma matriz de rotação, em torno da direção do veículo, à velocidade.

O *slip angle* β é estimado como a divisão da velocidade lateral pela longitudinal. Esta estimação é uma simplificação, pois o β real resulta da tangente dessa divisão. No entanto, devido aos valores reduzidos da velocidade lateral comparativamente à longitudinal, a divisão dá um valor muito semelhante ao da tangente da divisão, considerando-se esta simplificação aceitável. Esta simplificação tem a vantagem de facilitar o cálculo do Jacobiano, evitando-se também realizar as computacionalmente exigentes operações trigonométricas.

$$f(X_k, U_k) = \begin{cases} \frac{1}{m} \left((U_k(1) + U_k(2)) \cos(U_k(5)) + (U_k(3) + U_k(4)) - X_k(4) \sin(U_k(5)) \right. \\ \quad \left. - F_{DRAG} \right) + X_k(3)X_k(2) \\ \frac{1}{m} \left((U_k(1) + U_k(2)) \sin(U_k(5)) + (X_k(5)) + (X_k(4)) \cos(U_k(5)) \right) \\ \quad - X_k(3)X_k(1) \\ \frac{1}{I_z} \left(U_k(4) - U_k(3) + (U_k(2) - U_k(1)) \cos(U_k(5)) \right) \frac{l_s}{2} - (X_k(5)) l_r \\ \quad + \left((X_k(4)) \cos(U_k(5)) - (U_k(1) + U_k(2)) \sin(U_k(5)) \right) l_f \\ 0 \\ 0 \\ X_k(3) \\ X_k(1) \cos(X_k(6)) - X_k(2) \sin(X_k(6)) \\ X_k(1) \sin(X_k(6)) + X_k(2) \cos(X_k(6)) \end{cases} \quad (3.18)$$

assumindo-se

$$f_{k_{TOTAL}} f(X_k, U_k) = f_{k_{IMU}} f(X_k, U_k) = f(X_k, U_k) \quad (3.19)$$

Como no algoritmo proposto 1, este vai ter um conjunto de matrizes definidos de forma diferente para cada filtro.

A medição dos dispositivos de medição é definida, para o EKF_{TOTAL}, como (3.20), e para o EKF_{IMU} como (3.21):

$$h_{k_{TOTAL}}(X_k, U_k) = \begin{cases} X_k(1) \\ X_k(2) \\ \frac{1}{m} \left((U_k(1) + U_k(2)) \cos(U_k(5)) + (U_k(3) + U_k(4)) - X_k(4) \sin(U_k(5)) \right. \\ \quad \left. - F_{DRAG} \right) \\ \frac{1}{m} \left((U_k(1) + U_k(2)) \sin(U_k(5)) + X_k(5) + X_k(4) \cos(U_k(5)) \right) \\ X_k(6) \\ X_k(3) \\ X_k(2) \\ X_k(1) \\ X_k(7) \\ X_k(8) \end{cases} \quad (3.20)$$

$$h_{kIMU}(X_k, U_k) = \begin{cases} 0 \\ 0 \\ \frac{1}{m} \left((U_k(1) + U_k(2)) \cos(U_k(5)) + (U_k(3) + U_k(4)) - X_k(4) \sin(U_k(5)) \right. \\ \quad \left. - F_{DRAG} \right) \\ \frac{1}{m} \left((U_k(1) + U_k(2)) \sin(U_k(5)) + X_k(5) + X_k(4) \cos(U_k(5)) \right) \\ 0 \\ X_k(3) \\ \frac{X_k(2)}{X_k(1)} \\ 0 \\ 0 \end{cases} \quad (3.21)$$

A matriz Jacobiano F_k é definida da seguinte forma:

$$F_k = \begin{bmatrix} 0 & X_k(3) & X_k(2) & -\frac{\sin(U_k(5))}{m} & 0 & 0 & 0 & 0 \\ -X_k(3) & 0 & -X_k(1) & \frac{\cos(U_k(5))}{m} & \frac{1}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{l_f \cos(U_k(5))}{I_z} & \frac{l_r}{I_z} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \cos(X_k(6)) & -\sin(X_k(6)) & 0 & 0 & 0 & A_1 & 0 & 0 \\ \sin(X_k(6)) & \cos(X_k(6)) & 0 & 0 & 0 & A_2 & 0 & 0 \end{bmatrix}$$

onde

$$A_1 = -X_k(1) \sin(X_k(6)) - X_k(2) \cos(X_k(6))$$

$$A_2 = X_k(1) \cos(X_k(6)) - X_k(2) \sin(X_k(6))$$

Define-se:

$$F_{kTOTAL} = F_{kIMU} = F_k$$

As matrizes Jacobiano $H_{k_{TOTAL}}$ e $H_{k_{IMU}}$ são definidas, respetivamente, como:

$$H_{k_{TOTAL}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\sin(U_k(5))}{m} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\cos(U_k(5))}{m} & \frac{1}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \frac{-X_k(2)}{X_k(1)^2} & \frac{1}{X_k(1)} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_{k_{IMU}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\sin(U_k(5))}{m} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\cos(U_k(5))}{m} & \frac{1}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

As matrizes Jacobiano refletem as diferenças da matriz $h(X_k, U_k)$ para cada filtro e, novamente, a diferença entre elas resume-se à não presença das medições do GPS no EKF_{IMU}.

3.2.3 Discussão

O algoritmo proposto 1, que implementa um duplo filtro, permite a utilização de dispositivos de medição a diferentes frequências, evitando que a execução do filtro fique limitada à frequência de aquisição mais baixa, o que permite uma melhor linearização do modelo e um melhor funcionamento do EKF. Este algoritmo permite a correção da estimação inercial com a medição GPS quando esta fica disponível, evitando que se acumule erro do dispositivo de medição inercial e se verifique o *drift* provocado pelo erro.

Na algoritmo 2, introduz-se mais redundância e um vetor de estado maior, estimando-se mais variáveis. O objetivo não é obter uma performance superior, mas apenas mais informação de estado e mais redundância. No entanto, introduz-se também a componente de *drag*, que não estava inserida no modelo e que é determinante para descrever o movimento correto do veículo. O

resultado é um filtro versátil, onde é possível definir uma configuração única capaz de obter bons resultados para as manobras de teste.

Outras opções, dentro do mesmo tipo de algoritmo, para concretizar a fusão sensorial são o Unscented Kalman Filter (UKF) e um EKF de 2ª ordem. Não há nenhum algoritmo que resolva, totalmente, o problema da não linearidade, mas é possível conseguir uma performance superior em alguns casos.

O UKF aplica uma Unscented Transform (UT), que em vez de propagar a aproximação da covariância por um modelo linear de 1ª ordem, calcula um conjunto mínimo de pontos (pontos sigma) que capturam, com alta fidelidade, a média e covariância do sistema, que são posteriormente propagados pelo modelo não linear, evitando o complexo passo de linearização.

A vantagem deste filtro é o cálculo preciso da média e da covariância até à 3ª ordem (de uma expansão da série de Taylor), evitando o cálculo de Jacobianos, que podem ser operações bastante complexas de realizar e um desafio dependendo da complexidade do modelo não linear [52]. Este aumento de precisão é conseguido sem, teoricamente, um aumento de complexidade nem de tempo computacional, no entanto não obtém resultados significativamente melhores para sistemas com uma dinâmica não linear relativamente reduzida. No entanto, na prática, verifica-se que o UKF tem um tempo de execução igual ou superior ao EKF [55].

O Extended Kalman Filter de 2ª ordem (EKF2) é uma melhoria em relação ao EKF de 1ª ordem em termos de precisão mas, normalmente, mais pesado computacionalmente. O facto de realizar uma aproximação de 2ª ordem obriga a calcular mais um conjunto de Jacobianos, aumentando a complexidade do filtro e o número de cálculos a realizar. Apenas justifica a sua implementação em problemas que o EKF normal não consegue bons resultados. No entanto, é possível obter um EKF2 sem recorrer ao cálculo de Jacobianos, como demonstrado em [56], sendo uma aplicação bastante mais eficiente e obtendo uma estimação mais precisa do que o UKF.

Capítulo 4

Implementação do Algoritmo

Com o algoritmo desenvolvido, neste capítulo estuda-se a aplicação do mesmo a um caso prático, realizando-se um levantamento dos requisitos do sistema. Após este levantamento, analisa-se e escolhe-se a arquitetura e o hardware capaz de concretizar o algoritmo e de cumprir os requerimentos. É efetuado um estudo dos dispositivos de medição escolhidos, com o objetivo de identificar o erro e reduzir o seu impacto na estimação de estado, e identifica-se os protocolos de comunicação mais indicados para esta aplicação. Por fim, discute-se como será conseguida a implementação do algoritmo.

4.1 Requisitos do Sistema

Pretende-se estimar vários parâmetros do sistema, nomeadamente a velocidade longitudinal, lateral e angular, e as forças laterais que atuam nos eixos. É necessário ter acesso à aceleração, rotação, e velocidade do veículo a partir de dispositivos de medição. Na última versão do filtro, utiliza-se também a direção e posição do veículo para se estimar mais variáveis de estado, como a direção do veículo e o *side slip*.

Tendo em conta a dinâmica típica de um veículo de estrada, define-se que a aceleração não ultrapassará os $2g$ em qualquer direção. Define-se, também, que a velocidade angular não ultrapassará os $90^\circ/s$. A velocidade não é uma condicionante do sistema, ficando a velocidade medida e estimada restringida pelos limites operacionais do GPS, no valor de $500m/s$. O sistema funcionará durante curtos períodos de tempo, esperando-se uma distância percorrida de $1km$ no máximo.

Sendo este um sistema de posicionamento de um veículo, deverá estar preparado para comunicar com um sistema central de controlo de um veículo, sendo o sistema aqui desenvolvido um sub-sistema desse. Será necessário estabelecer comunicação com esse sistema e enviar e receber informação.

A frequência de trabalho do sistema que se pretende é $100Hz$. A comunicação com o sistema principal, a aquisição dos dispositivos de medição e a execução do algoritmo deve ser executada à frequência definida.

4.2 Hardware

4.2.1 Arquitetura do Hardware

O sistema desenvolvido será um sistema centralizado, com recurso a informação complementar. Serão utilizados dois dispositivos de medição, um IMU e um GPS. O IMU, constituído por acelerómetro, giroscópio e magnetómetro de 3 eixos, é um dispositivo MEMS que fornecerá a aceleração (e velocidade e posição através de integração), rotação e direção do sistema, obtendo-se assim a localização relativa do sistema. O recetor GPS fornece as coordenadas em latitude e longitude, obtendo-se a localização absoluta do mesmo. Fornece, também, a velocidade e direção com que o recetor se desloca. O IMU é configurável em termos de largura de banda, podendo ser mais ou menos sensível a vibrações.

Será usado um microprocessador que executará o algoritmo de fusão sensorial, usando a informação dos dispositivos de medição como entrada do algoritmo. Como o microprocessador terá que comunicar com os dispositivos para receber a informação destes, será necessário implementar um protocolo de comunicação que suporte vários componentes e que permita comunicações master/slave, para o microprocessador ter um controlo maior sobre o processo.

O objetivo é combinar dois tipos de informação de localização (relativa e absoluta) de forma a obter a maior precisão possível. A localização absoluta, com uma frequência de aquisição mais baixa (10Hz), tem como objetivo corrigir possíveis erros acumulados dos dispositivos de medição inerciais, que fornecem localização relativa a uma frequência superior.

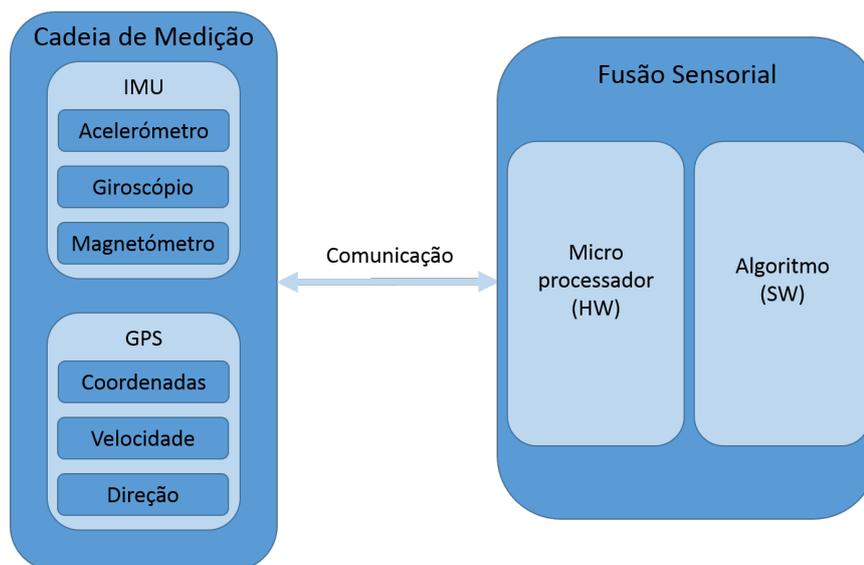


Figura 4.1: Representação simplificada do sistema a implementar

4.2.2 Dimensionamento do Hardware

A aceleração e velocidade angular do sistema são obtidos com recurso a um IMU, tipicamente compostas por um acelerómetro e giroscópio. Para obter a direção do sistema, é necessário um magnetómetro (que é uma bússola magnética). A velocidade e a posição do sistema são obtidas com um recetor GPS. O recetor GPS faculta a direção do movimento do sistema, que difere da direção do sistema físico dada pelo magnetómetro .

O IMU escolhido foi o MPU-9250, produzido pela InvenSense. É um dispositivo de medição inercial de 9 eixos, constituído por um acelerómetro, giroscópio e magnetómetro, facultando acesso às medições referidas anteriormente. Este IMU é a junção de outro dispositivo de medição com acelerómetro, giroscópio e um processador digital para algoritmos complexos (DMP), o MPU-6050, com um magnetómetro, o AK8963. Possui um termómetro, pois a temperatura pode influenciar a sensibilidade do dispositivo de medição. A InvenSense produz este IMU numa *breakout board*, estando já preparada para comunicar por I²C.

Este IMU tem uma gama de medições configurável. O acelerómetro permite escolher entre $\pm 2g$ e $\pm 16g$, e o giroscópio entre ± 250 até $\pm 2000^\circ/s$. É possível, também, ativar um filtro passa a baixo com diferentes configurações. O magnetómetro não tem configuração possível em termos de gama nem de filtro, tendo uma única gama de $\pm 4912\mu T$. O magnetómetro mede a força do campo magnético (densidade do fluxo magnético), ou seja, é medida termos de linhas de força por área, que é tipicamente expressa em gauss (G). $1T$ equivale a $10000G$ e o campo magnético terrestre é, tipicamente, medido em mG , sendo que este valor varia entre 250 e $650mG$. É possível representar a gama do magnetómetro em $\pm 49120mG$, confirmando-se a sua capacidade de medir o campo magnético que se verifica na Terra.

O recetor GPS usado é o Neo-6M, fabricado pela u-blox, juntamente com um módulo de navegação, de nome i2c-gps-nav. Este módulo, uma placa com um microprocessador AVR de 16MHz que estabelece uma comunicação série com o GPS, processa as mensagens do tipo UBX (protocolo proprietário da u-blox) fornecidas pelo GPS e converte-as para valores decimais. O módulo disponibiliza a informação em decimal por I²C, libertando o microprocessador central da comunicação com o GPS e da conversão da informação recebida. O código deste módulo é open-source e configurável, podendo-se alterar algumas funcionalidades, como a frequência de aquisição do GPS.

O módulo aplica um filtro à posição recebida do GPS, onde calcula uma média das últimas 5 posições recebidas e prevê a próxima posição com base nas velocidades em x e y , para compensar o atraso do GPS . O recetor, numa situação estática, é capaz de obter uma precisão de posição menor a $2,5m$, de acordo com a *datasheet*, e é algo verificado em teste. No entanto, em movimento, a precisão da medição de posição deteriora-se devido a variados fatores, como foi analisado no capítulo da revisão bibliográfica 2.

O GPS também fornece a velocidade e a direção do movimento, medições que necessitam do recetor em movimento para serem precisas. O recetor tem um *threshold* definido a partir do qual congela o cálculo da direção, se a velocidade for suficientemente baixa (aproximadamente

0,5m/s). Quando o recetor começa o movimento e ultrapassa este *threshold*, volta a calcular a direção. Devido a esta particularidade, não é possível recolher medições de velocidade e direção com o recetor estático, o que impede a análise e caracterização destas medições da mesma forma que se analisa a posição ou as medições do IMU. A u-blox indica uma precisão de 0,1m/s e 0,5° na *datasheet* do produto, mas indica que foram obtidos com o recetor numa posição estática, ou seja, esses valores podem não se verificar numa situação dinâmica.

Este recetor GPS é caracterizado por um rápido tempo de fix (comunicação estabilizada com satélites), após ligar. É necessário considerar três tipos de início de funcionamento: o *cold start*, *warm start* e *hot start* [57]. No *cold start*, o GPS descarta toda a informação que armazenou na última utilização, obrigando a que o GPS procure, novamente, os satélites sem informação prévia. No *warm start*, o GPS conhece a última posição calculada e tem a informação sobre o estado e tempo dos satélites em memória, mas não sabe com que satélites estabeleceu o *fix* na última utilização, tendo que estabelecer essa comunicação. No *hot start*, o GPS tem toda a informação da última utilização, tentando comunicar com os mesmos satélites com que conseguiu o *fix* da última vez. O *hot start* apenas acontece quando se desliga e liga o GPS num curto espaço de tempo (minutos) e perto da mesma posição em que foi desligado. Os tempos de *start* para este GPS são de 26 segundos para o *cold* e *warm start*, e de 1 segundo para o *hot start*.

O par GPS+módulo é popular num projeto open-source de controlo de drones, o MultiWii, devido às múltiplas funcionalidades que disponibiliza além da conversão das mensagens do GPS. Fornece um controlador PID para manter a posição ou movimentar o drone, e permite definir destinos com base em valores de latitude e longitude. No entanto, estas funcionalidades não serão utilizadas nesta dissertação.

De forma a acelerar a comunicação I²C e para ser feita em apenas um pedido, foi alterada a ordem de envio dos registers, para não se perder tempo em receber informação do GPS que não é relevante.

O componente principal, o processador, será um microprocessador do tipo Arduino. Inicialmente considerou-se a placa Arduino Nano, por ser um componente compacto com uma comunidade muito ativa, havendo vários recursos disponíveis online e tendo a capacidade de implementar os principais tipos de comunicação com facilidade. Este arduino possui um microcontrolador ATmega328 com um *clock speed* de 16MHz, tem 32KB de memória flash e 2KB de SRAM.

O algoritmo (versão 1) desenvolvido em Matlab foi re-escrito em código C, para ser executado pelo microprocessador. Testou-se o código com um script em Python (Anexo A) para enviar, pela porta série, os valores de teste das medições dos dispositivos de medição e do vetor controlo, e para receber os cálculos do filtro. Neste teste foi possível perceber que o algoritmo demorava demasiado tempo a ser executado (< 100Hz), o que tornaria a execução do programa excessivamente lenta com a comunicação ainda por introduzir.

Escolheu-se, como alternativa ao Arduino Nano, o Teensy 3.2. Considerou-se esta placa devido à sua facilidade de utilização, ao mesmo nível do Arduino, e devido à sua compatibilidade com bibliotecas e programas desenvolvidos para Arduino. Esta placa contém um processador ARM de 32-bits Cortex-M4, com um clock speed de 72MHz (suscetível a overclock). Em termos

de memória, contém 256KB de Flash e 64 KB de RAM, o que, comparando com o Arduino Nano, é um upgrade significativo, sem um aumento físico de espaço. Enquanto que com o Arduino havia a preocupação de não se poder ultrapassar a memória disponível, com o Teensy ganhou-se mais liberdade. Com este novo microprocessador, será possível obter um aumento de velocidade no mínimo 4x superior, devido a uma velocidade de processador 4,5x superior à do Arduino e, também, a uma arquitetura de processador diferente (ARM de 32-bits do Teensy contra AVR de 8-bits do Arduino).

O código desenvolvido para o Arduino foi alterado para funcionar com o Teensy. Foi necessário fazer algumas adaptações e usar uma biblioteca matemática para realizar cálculos matriciais, pois a usada para o Arduino não era compatível. Realizou-se o teste, com o script Python, com o Teensy, e observou-se uma melhoria significativa em termos de velocidade, com o algoritmo a executar entre 9-10x mais rápido.

Por fim, utiliza-se um cartão de memória para se realizar um datalogging de certa informação do sistema. Simplesmente escolheu-se uma shield de cartão microSD para Arduino, ligando-a ao Teensy e realizando o datalog em código no programa principal.

4.3 Caracterização dos Dispositivos de Medição

Todos os dispositivos de medição são afetados, de alguma forma, por erro que influencia negativamente as medições do sistema. Os de melhor qualidade e, conseqüentemente, mais caros, conseguem reduzir esse erro, tendo impacto em aplicações práticas apenas após um longo período de utilização, como já foi analisado no Capítulo 2. Identificar e corrigir o erro presente nos dispositivos é essencial para a utilização dos mesmos.

Para identificar o erro, recorreu-se ao método Variância de Allan, que consiste na análise de um conjunto de dados no domínio do tempo para medir a frequência de estabilidade em osciladores [58]. Este método pode ser utilizado para identificar o ruído, e as possíveis causas do mesmo, de um IMU, sendo um método simples de compreender e de computar. Os cinco erros básicos são descritos como *angle* e *velocity random walk*, *rate random walk* (passeio aleatório), instabilidade de bias, ruído de quantização e *drift rate ramp*. Os que afetam principalmente os MEMS são a instabilidade de bias e o *random walk*. Este método é apenas válido para o acelerómetro e giroscópio, uma vez que análise realizada é sobre a integração da informação obtida do dispositivo de medição.

Os autores El-Sheimy et al. em [7] aplicam, com sucesso, este método para estudar o erro de dispositivos de medição inercial, salientando que os resultados obtidos estão de acordo com o que os fabricantes reivindicam, legitimando este método neste tipo de estudo para o IMU.

4.3.1 Metodologia

Definindo-se o método [7] em função do output de aceleração e de velocidade angular do dispositivo de medição e assumindo N amostras consecutivas, com uma tempo de aquisição t_0 ,

formam-se grupos de n amostras consecutivas (com $n < N/2$, para haver no mínimo 2 grupos), como está representado na Figura 4.2:

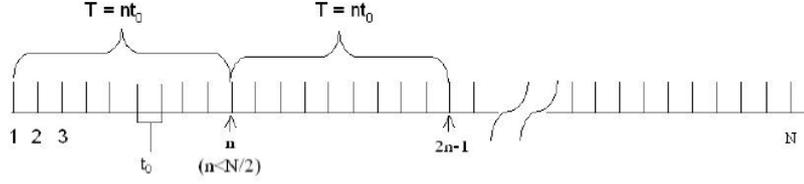


Figura 4.2: Esquema da estrutura de dados usada na Variância de Allan [7]

Para cada grupo associa-se um tempo, T , com valor nt_0 , obtendo uma média do grupo a partir da Equação (4.1):

$$\bar{\Omega}_k(T) = \frac{1}{T} \int_{t_k}^{t_k+T} \Omega(t) dt \quad (4.1)$$

onde $\bar{\Omega}_k(T)$ representa a média do grupo de tempo considerado para o output do dispositivo de medição, começando na amostra k com n amostras. Logicamente, a média do seguinte grupo de tempo é (4.2):

$$\bar{\Omega}_{next}(T) = \frac{1}{T} \int_{t_{k+1}}^{t_{k+1}+T} \Omega(t) dt \quad (4.2)$$

onde $t_{k+1} = t_k + T$.

Aplicando a Equação (4.3):

$$\xi_{k+1,k} = \bar{\Omega}_{next}(T) - \bar{\Omega}_k(T) \quad (4.3)$$

obtém-se um conjunto de variáveis aleatórias, interessando a variância entre todos os grupos de tempo formados a partir do mesmo conjunto de dados. A variância de Allan, de tamanho T , pode ser definida pela Equação (4.4) ($\langle \rangle$ representa a média):

$$\sigma^2(T) = \frac{1}{2} \left\langle [\bar{\Omega}_{next}(T) - \bar{\Omega}_k(T)]^2 \right\rangle \quad (4.4)$$

resultando em (4.5):

$$\sigma^2(T) = \frac{1}{2(N-2n)} \sum_{k=1}^{N-2n} [\bar{\Omega}_{next}(T) - \bar{\Omega}_k(T)]^2 \quad (4.5)$$

A Equação (4.5) representa a estimacão de uma quantidade $\sigma^2(T)$ calculada a partir de um conjunto de dados de tamanho N , onde um número finito de grupos de tempo com um tamanho fixo são formados.

A variância de Allan pode, também, ser definida em termos de output de velocidade e ângulo, em vez de aceleração e velocidade angular. É necessário integrar os valores dados pelo dispositivo

de medição, expresso na Equação (4.6):

$$\theta(t) = \int^t \Omega(t) dt \quad (4.6)$$

As medições de velocidade ou ângulo são feitas em tempo discreto, dados por $t = kt_0, k = 1, 2, 3, \dots, N$. Simplifica-se a notação escrevendo $\theta_k = \theta(kt_0)$, e as Equações (4.1) e (4.2) podem ser redefinidas como a Equação (4.7):

$$\bar{\Omega}_k(T) = \frac{\theta_{k+n} - \theta_k}{T}, \quad \bar{\Omega}_{next}(T) = \frac{\theta_{k+2n} - \theta_{k+n}}{T} \quad (4.7)$$

A Equação (4.5), a variância de Allan, passa a ser estimada pela Equação (4.8):

$$\sigma^2(T) = \frac{1}{2T^2(N-2n)} \sum_{k=1}^{N-2n} (\theta_{k+2n} - 2\theta_{k+n} + \theta_k)^2 \quad (4.8)$$

Por fim, calcula-se o desvio de Allan a partir da Equação (4.9):

$$\sigma(T) = \sqrt{\sigma^2(T)} \quad (4.9)$$

usando-se este resultado para caracterizar o ruído do IMU.

A Variância de Allan é uma medição da estabilidade dos dispositivos de medição, significando que está relacionada com as propriedades estatísticas do processo aleatório intrínseco, que afetam as medições. O resultado obtido aplicando o método referido está relacionado com a densidade espectral (*power spectral density* - PSD) dos ruídos da informação original. Essa relação é descrita pela Equação (4.10):

$$\sigma^2(T) = 4 \int_0^{\infty} S_{\Omega}(f) \frac{\sin^4(\pi f T)}{(\pi f T)^2} df \quad (4.10)$$

onde $S_{\Omega}(f)$ é a PSD do processo aleatório $\Omega(T)$. A PSD de qualquer processo aleatório pode ser substituída no integral, obtendo-se uma expressão para a Variância de Allan $\sigma^2(T)$ em função do comprimento do grupo de tempo.

Pode-se, então, identificar e quantificar o ruído e as suas fontes em medições inerciais, a partir de um gráfico log-log do desvio de Allan $\sigma(T)$ em função de T . Analisando este gráfico, pode-se tirar conclusões sobre o ruído a partir dos declives que existam, que serão identificados e analisados de seguida. Ruídos diferentes têm perfis de PSD diferentes: ruídos de alta frequência aparecerão no 1º declive do gráfico, e os de baixa frequência no 2º declive.

4.3.2 Erro

Além da Variância de Allan, considerou-se, também, o desvio padrão das medições dos dispositivos de medição. Uma vez que a Variância de Allan caracteriza os dispositivos de medição em função da velocidade e ângulo, o desvio padrão será o parâmetro utilizado para modelizar o ruído branco das medições de aceleração, velocidade angular e campo magnético.

De forma a obter os dados necessários para esta análise recolheu-se 1 milhão de amostras a $1kHz$, que foram guardadas num cartão SD. Recolheu-se 6 milhões de amostras, para os 3 eixos do acelerómetro e do giroscópio. Para o magnetómetro, recolheu-se o mesmo número de amostras para os 3 eixos, mas a uma frequência mais baixa ($100Hz$) resultando num total de 3 milhões de amostras. Nas medições do eixo z do acelerómetro, é necessário subtrair o valor de $1g$ às medições antes de calcular o bias, devido ao alinhamento do eixo z com a força da gravidade, medindo-a, resultando num bias fixo. Calculando-se a média e o desvio padrão das leituras do acelerómetro, giroscópio e magnetómetro obtém-se a Tabela 4.1:

Tabela 4.1: Bias do IMU

IMU		Bias	Desvio padrão
Acelerómetro (m/s^2)	Eixo X	-0,01078	0,01997
	Eixo Y	0,01130	0,02022
	Eixo Z	-0,01663	0,03184
Giroscópio ($^\circ/s$)	Eixo X	-2,954	0,1003
	Eixo Y	0,7593	0,1033
	Eixo Z	-0,1061	0,1137
Magnetómetro (mG)	Eixo X	405,80	7,711
	Eixo Y	50,626	7,714
	Eixo Z	-256,81	7,618

Aplicou-se o método Variância de Allan a cada conjunto de dados, recorrendo-se à implementação Matlab de [58]. Os resultados do acelerómetro e do giroscópio, como têm o mesmo número de amostras e foram amostrados à frequência, encontram-se na mesma Figura 4.3:

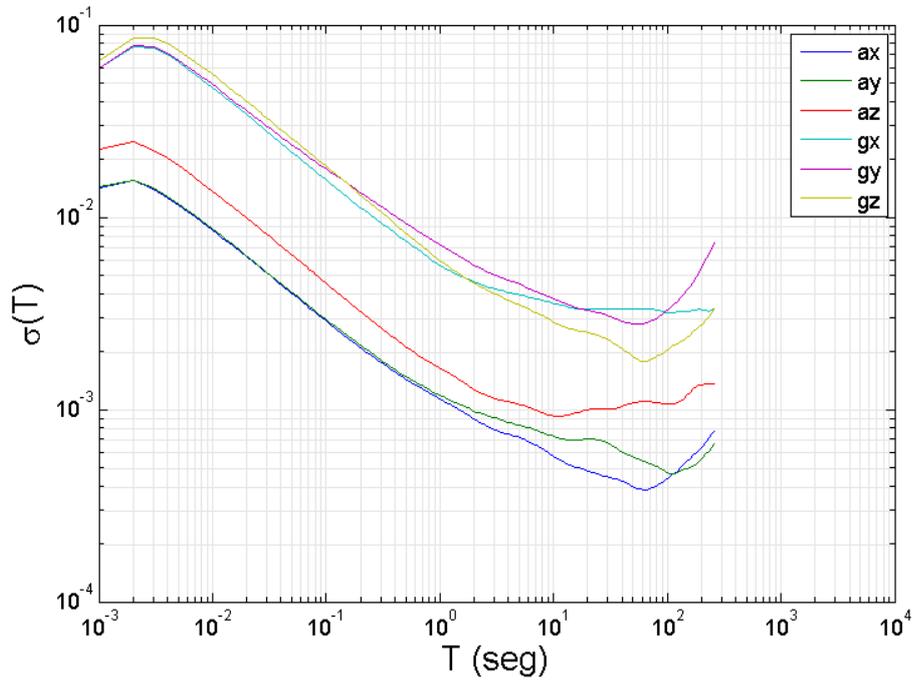


Figura 4.3: Desvio de Allan para os 3 eixos do acelerómetro e do giroscópio

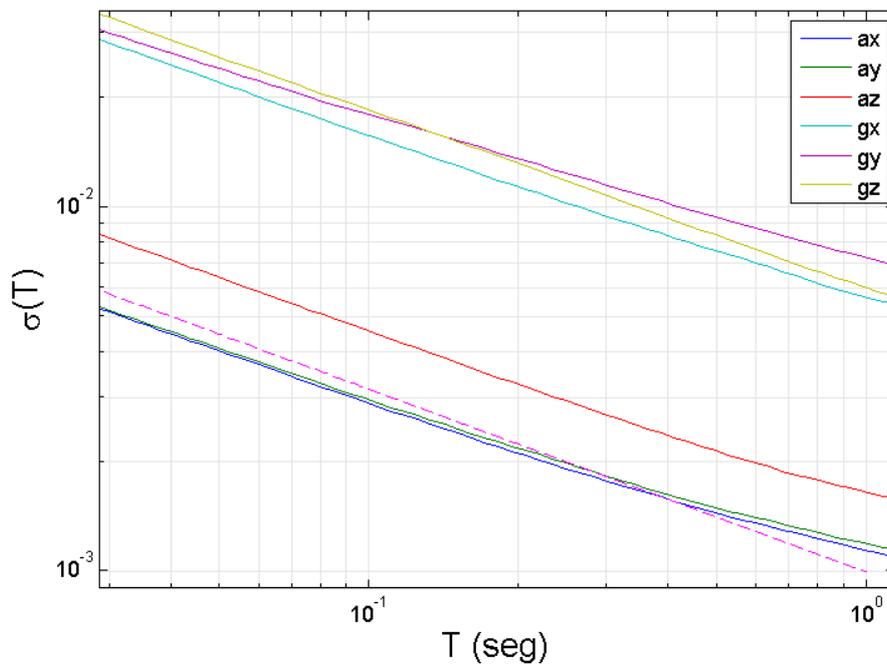


Figura 4.4: Ampliação do Desvio de Allan para os 3 eixos do acelerómetro e do giroscópio, com adição de uma reta (a magenta) de declive $-0,5$

Observa-se um padrão muito semelhante em todas as linhas da Figura 4.3, onde o valor do desvio de Allan tem um pico praticamente no início do gráfico, diminuindo até atingir um valor mínimo. Há algumas diferenças entre eixos, como o a_z ter valores ligeiramente superiores ao a_x e a_y , que podem ser explicados por uma diferença de sensibilidade individual na medição, ou pelo próprio alinhamento dos eixos que, apesar dos IMU tipicamente compensarem estes erros, ainda se pode observar alguma variação.

Analisando a Figura 4.3, identifica-se uma recta com declive $-0,5$ ou muito perto desse valor para todas as linhas. Na Figura 4.4 realizou-se uma ampliação do gráfico e acrescentou-se uma linha magenta a tracejado com declive $-0,5$, de equação $y = kx^m$, com $m = -0,5$ e $k = 0,001$, onde m é o declive da recta e k é o valor de y em que a linha intercepta o eixo x no ponto $x = 1$, resultando numa linha retilínea num gráfico log-log. Esta linha foi adicionada com o intuito de demonstrar o declive do gráfico.

O declive $-0,5$ está associado ao erro *angle random walk* (ARW), para o giroscópio, e *velocity random walk* (VRW), para o acelerómetro. São expressos em $^\circ/\sqrt{Hz}$ e $(m/s)/\sqrt{Hz}$, respetivamente, e é caracterizado pela integração do ruído branco do output do dispositivo de medição. Este erro é encontrado no ponto $T = 1$ (10^0s) de cada linha [7]. Multiplicando esse valor por $\sqrt{1/3600}$, que corresponde a uma hora, obtém-se a ARW/VRW por hora, cujos valores estão na Tabela 4.2.

Tabela 4.2: ARW do giroscópio e VRW do acelerómetro

IMU	VRW/ARW	
Acelerómetro ($(m/s)/\sqrt{H}$)	Eixo X	0,001125
	Eixo Y	0,001178
	Eixo Z	0,001626
Giroscópio ($^\circ/\sqrt{H}$)	Eixo X	0,005550
	Eixo Y	0,007128
	Eixo Z	0,005866

Outro caso comum em todas as linhas da Figura 4.3 é um vale, onde o declive aumenta lentamente, passando de negativo para positivo. O valor mínimo do gráfico, que acontece neste vale, está relacionado com a instabilidade de bias [7], que é dado em $(m/s)/s$ para o acelerómetro e $^\circ/s$ para o giroscópio. É possível obter este erro por hora, em vez de por segundo, multiplicando o valor obtido por 3600. Este erro existe devido a eletrónica do sensor ou a outros componentes suscetíveis a oscilações aleatórias [7], e indica quanto o output do dispositivo de medição pode variar em função do tempo, a uma temperatura constante. Este bias é considerado bias dinâmico e, juntamente com o estático, constituem o bias total. Na Tabela 4.3, encontram-se os valores medidos para este erro.

desempenho deste recetor não diverge de um usado num *smartphone*.

Um atributo do par GPS+Módulo utilizado é o ruído reduzido das medições. Apesar de haver um limite na precisão obtida, o GPS é capaz de acompanhar o movimento do veículo sem haver saltos de posição, sendo possível identificar o percurso que o veículo fez apesar da medição sofrer desvios em relação à realidade.

4.3.3 Correção do Erro

O ARW/VRW não será passivo de correção devido à natureza do filtro usado, o EKF. O filtro recorre, com frequência, a medições de velocidade obtidas pelo GPS que corrigem possíveis discrepâncias entre os valores calculados pelo modelo e pelo que se observa nas medições, quer em termos de velocidade ou aceleração, eliminando o *drift* provocado pela integração das medições inerciais.

Considerando que os valores da instabilidade de bias, que tem um impacto crescente com o período de funcionamento do dispositivo de medição, são na ordem dos $1 - 3(m/s)/h$ e $5 - 10^\circ/h$ e como este sistema será executado em curtos períodos de tempo, tomou-se a decisão de não se considerar este erro na modelização do dispositivo nem no desenvolvimento do filtro de Kalman, uma vez que o erro não atingirá um nível que afete os cálculos.

Na execução do filtro, irá considerar-se o desvio padrão das medições dos dispositivos de medição para modelizar o ruído branco do mesmo. O filtro, sendo um método probabilístico, é capaz de reduzir o efeito do ruído branco, necessitando apenas de conhecer um parâmetro que caracterize o ruído branco verificado no dispositivo de medição - o desvio padrão é um bom parâmetro para modelizar o ruído.

O erro estático presente nas medições é corrigido através de uma calibração. É possível obter o bias da medição calculando a média de um número razoável de amostras depois de se ligar o dispositivo de medição, bastando subtrair esse valor às medições realizadas para obter uma medição sem bias. Analisando as amostras recolhidas e referidas no ponto anterior, é possível calcular os valores representados na Tabela 4.1. No entanto, como o valor de bias varia de *run-to-run*, é necessário fazer esta operação sempre que se liga o dispositivo de medição.

No caso do magnetómetro, há uma componente de bias associada ao mau alinhamento do referencial com a origem, que não é corrigido pelo método referido anteriormente, pois um mau alinhamento significa que as medições são feitas em relação ao referencial mal alinhado. O processo para eliminar este erro passa por identificar o centro do referencial do magnetómetro em relação à origem e compensar esse desvio subtraindo-o às medições realizadas.

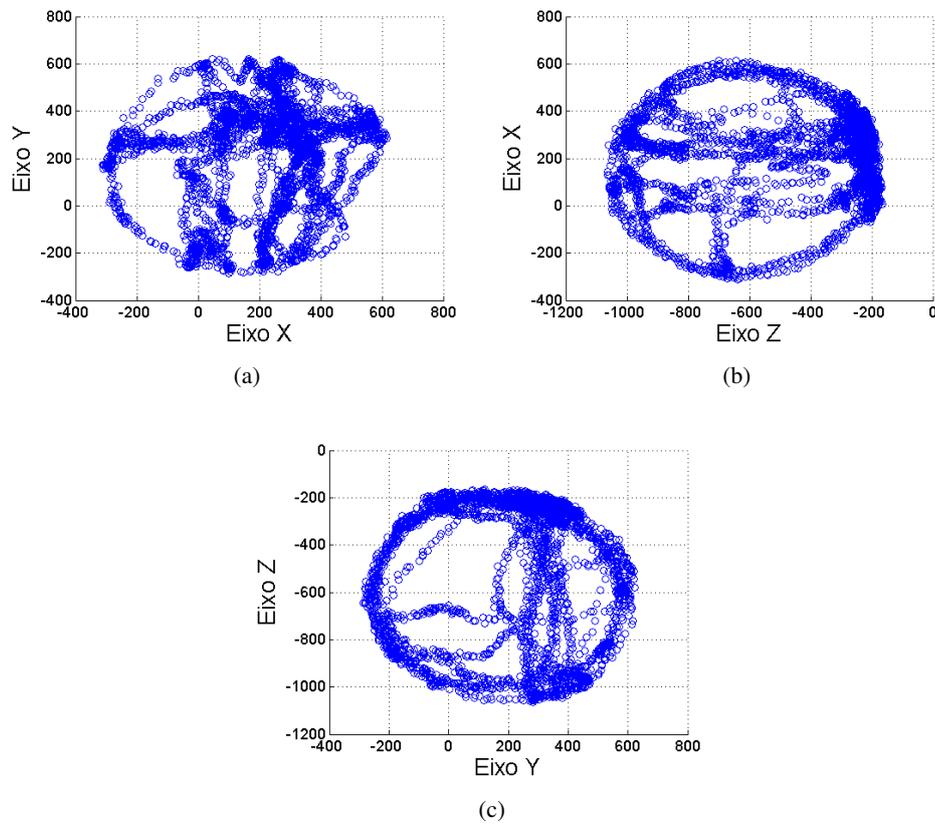


Figura 4.6: Gráfico eixo vs eixo das medições do magnetómetro: (a) eixo x vs y , (b) eixo z vs x , (c) eixo y vs z

O centro do referencial foi identificado rodando o dispositivo de medição em torno dos seus 3 eixos, enquanto se recolhia amostras, obtendo amostras em todas as direções do espaço 3D. É fundamental garantir que, ao rodar o dispositivo de medição, se cobre o máximo de direções possíveis, para obter uma maior confiança no cálculo do bias. Obteve-se as seguintes leituras, comparando-se todos os eixos entre eles na Figura 4.6.

O valores do magnetómetro amostrados são o output *raw*, não estando expressos em nenhuma unidade. É nítido que o centro de cada eixo está desviado do ponto 0, sendo necessário subtrair os valores da Tabela 4.4, obtidos realizando uma média entre o máximo e mínimo encontrado em cada eixo, para obter uma leitura centrada na origem nos 3 eixos.

Tabela 4.4: Bias fixo do IMU

IMU		Output raw
Magnetómetro	Eixo X	151,535
	Eixo Y	168,670
	Eixo Z	-614,920

Este método é iterável, sendo que se pode voltar a executar o processo, depois de se corrigir o bias com o resultado da iteração anterior, de forma a obter um referencial mais centrado na origem.

4.3.4 Transformadas

Apesar dos valores do IMU já estarem representados no CoG e por esse motivo não necessitarem de qualquer tipo de transformação, é necessário converter os valores obtidos do MPU-9250 para unidades SI. O IMU fornece valores de 16 bits para as três grandezas, sendo que o valor de output depende da sensibilidade definida. A sensibilidade depende da gama de leituras que se deseja obter e, considerando a gama mais curta possível, $\pm 2g$ e $\pm 250^\circ/s$, tem-se uma sensibilidade de $16384LSB/g$ para o acelerómetro e $131LSB/(^\circ/s)$ para o giroscópio. Esta é a configuração que também permite uma maior resolução das medições. O magnetómetro não é configurável em termos de sensibilidade, tendo uma gama fixa de $\pm 4912\mu T$ e sensibilidade de $15\mu T/LSB$.

Para obter os valores em g e $^\circ/seg$ para o acelerómetro e giroscópio, respetivamente, é necessário multiplicar o valor obtido pela sensibilidade e para os obter em unidades SI (m/s^2 e rad/seg), multiplica-se por 9,8 e $\pi/180$, respetivamente. Para o magnetómetro, é necessário multiplicar pela resolução, pela sensibilidade definida para cada eixo e subtrair o bias que é calculado manualmente. O valor efetivo do campo magnético não é relevante, apenas é importante que a escala entre os diferentes eixos seja a mesma, obtendo-se o ângulo da direção do veículo (DV) através de $\arctan(m_y, m_x)$.

Do GPS, obtém-se a velocidade do movimento, a direção do movimento (DM) e a posição absoluta. A velocidade é transmitida pelo módulo em m/s com uma resolução máxima de velocidade de $0,01m/s$ e de direção de $0,1^\circ$.

A velocidade longitudinal e lateral não são medidas diretamente, mas é possível obter a velocidade longitudinal e lateral do veículo através da velocidade dada pelo GPS e pelo *side slip* do veículo. O *side slip*, neste sistema, não é medido a partir de dispositivos de medição - mas é possível obtê-lo conjugando a DM com a DV através da Equação (4.11):

$$\beta = DV - DM \quad (4.11)$$

Conhecendo-se o *side slip*, obtém-se a velocidade longitudinal e lateral pela Equação (4.12):

$$v_x = v \cos(\beta), \quad v_y = v \sin(\beta) \quad (4.12)$$

onde v é a velocidade medida pelo GPS.

A posição é obtida numa *frame* global, o Sistema de Coordenadas Geodésico, em latitude e longitude. No entanto, de forma a poder ser utilizada pelo algoritmo, a posição ou a variação de posição deve estar em metros e expressa num referencial 2D, uma vez que não se pode utilizar a latitude e longitude diretamente para calcular distância. Uma possibilidade seria realizar a conversão pelo método de Haversine, que calcula a distância mais curta pela superfície terrestre de um ponto a outro. Este método considera a Terra como uma esfera e não uma elipse, tendo um erro,

no pior caso, de 0,55%, mas usualmente é menor do que 0,03%, obtendo uma precisão elevada para curtas distâncias [59]. A fórmula está exposta na Equação (4.13):

$$\begin{aligned} a &= \sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos(\varphi_1)\cos(\varphi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right) \\ c &= 2 \operatorname{atan2}(\sqrt{a}, \sqrt{1-a}) \\ d &= Rc \end{aligned} \quad (4.13)$$

onde λ_2 e λ_1 são a longitude de dois pontos terrestres, φ_2 e φ_1 a latitude de dois pontos terrestres, R o raio da Terra e d a distância do grande círculo entre dois pontos. Este método, no entanto, é muito dispendioso computacionalmente devido ao elevado número de operações trigonométricas realizadas.

Optou-se, então, por uma transformada de coordenadas através de uma projecção equiretangular [60]. Esta transformada projeta as linhas de latitude em linhas retas, paralelas e igualmente espaçadas entre elas, realizando a mesma transformação para os meridianos, necessitando apenas de um fator de escala. Esta projecção é conseguida através das Equações (4.14):

$$\begin{aligned} x &= R\lambda \cos(\varphi_0) \\ y &= R\varphi \end{aligned} \quad (4.14)$$

onde λ é a longitude, φ a latitude, φ_0 a latitude de origem, e x e y os eixos cartesianos que definem os pontos λ e φ num plano cartesiano 2D. φ_0 e $\cos(\varphi_0)$ são a latitude da origem e o fator de escala que tornam esta projecção válida para o respetivo par de linhas de latitude equidistantes acima e abaixo da linha do equador.

Considerando-se λ_0 e φ_0 como a longitude e latitude iniciais do sistema, que resultam em x_0 e y_0 através da equação anterior, e considerando-se que o ponto inicial x_0 e y_0 são o x_{IN} e y_{IN} do referencial origem da Figura 3.1, pode-se calcular a distância percorrida, em metros, de uma nova medição de latitude e longitude a partir da Equação (4.15):

$$d = \sqrt{(x - x_0)^2 + (y - y_0)^2} \quad (4.15)$$

onde x e y são as novas leituras de latitude (φ) e longitude (λ) projetadas pela Equação (4.14). A aplicação desta projecção provoca distorções locais, cuja magnitude varia com o local da origem. Se um determinado ponto se encontrar na linha do equador, o erro entre a latitude e longitude é constante, ou seja, não há distorção. Mas quanto mais longe do equador for a latitude de origem, maior a distorção nos eixos à medida que nos afastamos do ponto de origem, resultando num erro considerável e de cálculo razoavelmente complexo [61].

Este método pode ser utilizado para calcular distâncias entre locais a curtas distâncias, relativamente ao ponto de origem, assumindo-se uma boa precisão. Para testar a precisão desta projecção, consideraram-se distâncias até aproximadamente 5,5km. Comparou-se o erro obtido no cálculo de distância por esta projecção com o do método Haversine, na região do Porto, Portugal.

Compararam-se 12 pontos em relação a um ponto de origem, todos a distâncias diferentes e mais ou menos deslocados em latitude ou longitude, representados na Figura 4.7, que foram obtidos através do Google Maps. Na imagem não se encontram os pontos F, G, H e I, pois estão a uma distância demasiado curta do ponto de origem, o ponto M, aparecendo o ponto M sobreposto na imagem. Na Tabela 4.5 encontra-se o erro deste método, obtido recorrendo ao script apresentado no Anexo B.

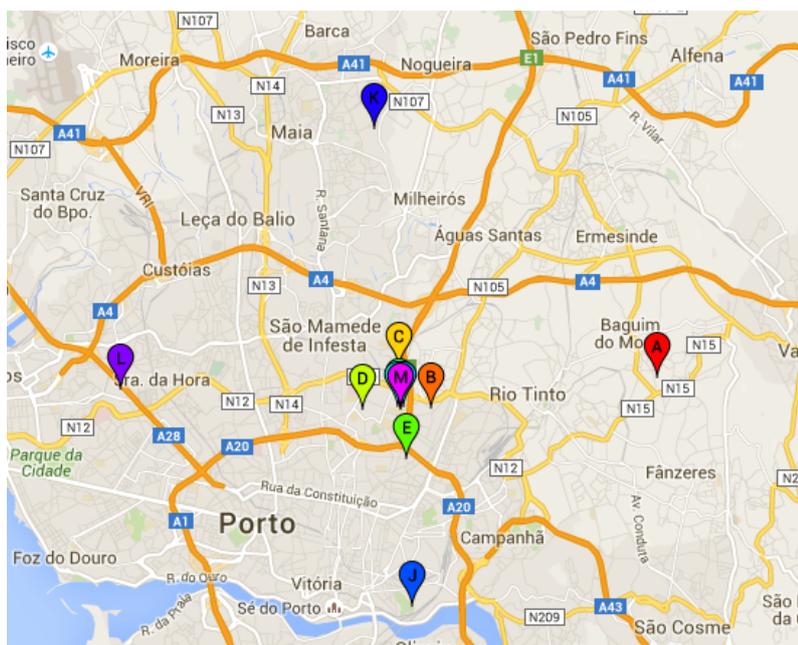


Figura 4.7: Mapa com os pontos definidos para calcular o erro da projeção

Tabela 4.5: Erro da projeção equiretangular

Ponto	Latitude	Longitude	Distância à origem (m) - M: (41,1785288;-8.5956001)		
			Haversine	Projeção equirectangular	Erro (%)
B	41,1784602	-8,5882187	617,813	617,812	0,0000523
C	41,1856305	-8,5958576	789,967	789,967	0,00000403
D	41,1781372	-8,6044407	741,171	741,169	0,000298
E	41,1693507	-8,5942268	1027,01	1027,01	0,0000877
F	41,1781533	-8,5955358	420,991	420,991	0,00000468
G	41,1791869	-8,5956860	735,297	735,297	0,00000480
H	41,1786217	-8,5962653	566,222	566,223	0,0000686
I	41,1786701	-8,5947418	735,311	735,312	0,000103
J	41,1429843	-8,5926819	3959,91	3959,91	0,000103
K	41,2282490	-8,6016083	5551,44	5551,45	0,000311
L	41,1817548	-8,6613464	5513,99	5514,13	0,00245
A	41,1838219	-8,5350037	5105,28	5105,48	0,00399

Verifica-se que o erro causado pela distorção da projeção é maior quando há um deslocamento em longitude. Com esta projeção, é possível obter uma precisão na ordem dos $0,2m$ relativamente a um método de cálculo de distâncias com precisão elevada, em distâncias até $5,5km$, assumindo-se esse erro desprezável.

4.4 Comunicação

De forma a integrar os dispositivos de medição com o micro-processador, escolheu-se um protocolo que permitisse a comunicação entre os vários aparelhos. Procurou-se utilizar o tipo de comunicação master-slave, com a possibilidade de existirem vários slaves, e também com a possibilidade de existir mais do que um master (multi-master). A distância considerada para a comunicação é uma distância muito curta (inferior a 1 metro). A velocidade de transmissão também é importante, tentando-se obter uma velocidade elevada mas sem sacrificar os outros pontos, que são prioritários.

O I²C usa dois sinais, o SCL (clock signal) e SDA (data signal), para estabelecer a comunicação entre instrumentos. O bus é, portanto, constituído por apenas 2 fios, onde se pode ligar vários aparelhos, que são comunicáveis a partir de um endereço único no bus. Apesar deste protocolo suportar endereços de 7 e 10 bits, considera-se apenas os de 7 bits, resultando em 128 endereços possíveis mas com alguns deles reservados, podendo-se ligar 112 instrumentos. O bus é do tipo coletor aberto, ou seja, normalmente está no seu estado HIGH, sendo que os aparelhos puxam os sinais para o estado LOW. O master que estabelece a comunicação controla o sinal de clock, a uma frequência definida em software que os slaves têm de ser capazes de utilizar, pois depende da frequência do processador, usando o SDA para transmitir os bytes ao slaves, resultando numa

comunicação simples. O I²C suporta multi-master, implementando lógica de arbitração e detecção de ocupação do bus de forma a impedir que um master comunique quando o bus está ocupado por outro master [8].

O I²C comunica por bytes de 8 bits, e necessita de enviar pelo menos dois bytes por comunicação: um com o endereço de 7 bits do slave + o bit de read/write, que será o destino da mensagem, e um byte de informação de 8 bits. Uma comunicação iniciada por um master terá um start bit, e cada byte enviado será seguido por um bit de *acknowledge* (ACK), que permite ao recetor indicar que recebeu o byte. O master envia um stop bit quando termina a comunicação. Na Figura 4.8 encontra-se a frame da comunicação do I²C com 2 bytes enviados – 1 pelo master, e outro pelo recetor (que pode ser o master ou o slave).

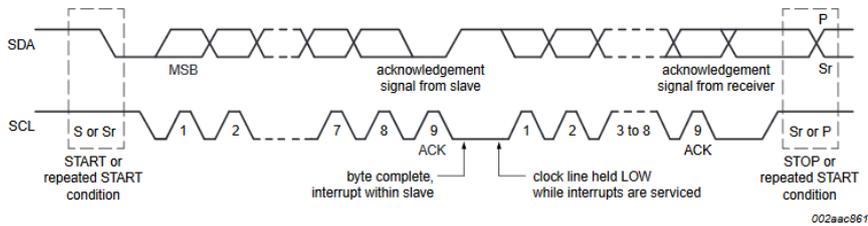


Figura 4.8: Exemplo de uma frame da comunicação por I²C [8]

O I²C permite três formas de comunicação. Um onde o master envia informação para o slave, começando com o byte do endereço do slave + bit de write (indicando que vai escrever), seguido de um certo número de bytes de informação, com o slave a mandar um ACK sempre que recebe um byte 4.9; um segundo onde o master passa para recetor imediatamente depois de enviar o primeiro byte – o master envia um byte com o endereço do slave + um read (indicando que vai ler), o slave responde com um ACK e o master e slave trocam de papéis, com o slave responsável por enviar o seguinte byte e com o master a enviar um $\overline{\text{ACK}}$ depois de receber o byte, enviando um stop byte quando quer parar de receber bytes. Existe, ainda, um terceiro modo, que é uma combinação dos dois anteriores: o master envia um certo número de bytes, como na primeira forma descrita, mas depois segue-se a escrita do slave, como na forma 2, demonstrada na Figura 4.10.

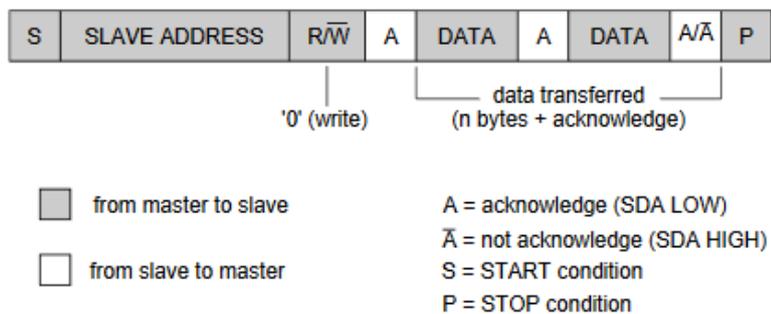


Figura 4.9: Exemplo de escrita por parte de um master no protocolo I²C [8]

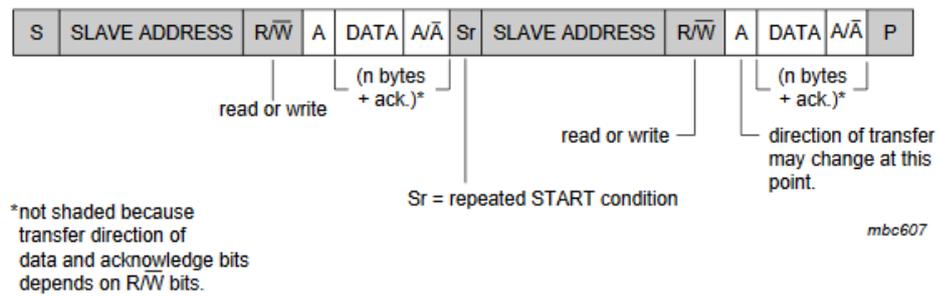


Figura 4.10: Exemplo de leitura por parte de um master no protocolo I²C [8]

Os dispositivos de medição usados necessitam de múltiplos endereços de memória para armazenar e disponibilizar toda a informação que medem; Consequentemente, o segundo modo de comunicação não será usado – é inútil pois, além de ser preciso enviar o endereço do slave, é também necessário enviar o endereço a que queremos aceder. Usa-se, portanto, o primeiro modo de comunicação, com o objetivo de configurar os dispositivos de medição, e usa-se o terceiro para pedir e receber informação dos mesmos, onde se envia dois bytes iniciais, seguido de um byte com um pedido de leitura, recebendo os bytes enviados por parte do slave até o master mandar o slave parar, com um $\overline{\text{ACK}}$ e um stop bit.

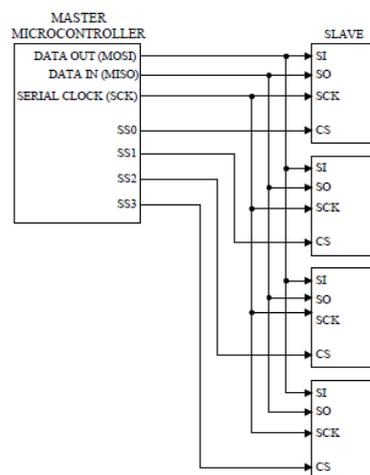


Figura 4.11: Comunicação multi-slave através do protocolo SPI [9]

SPI (*Serial Peripheral Interface*) é uma interface série síncrona muito comum para transmitir dados, desenvolvida pela Motorola. A arquitetura deste sistema de comunicação é master-slave, com a possibilidade de haver múltiplos slaves, sendo um sistema de comunicação full-duplex, permitindo o envio e receção de dados simultaneamente entre o master e slave. Este protocolo necessita de 4 fios para estabelecer comunicação entre um master e slave, e necessita de mais um fio e pino por cada slave ligado. Um dos fios, o Chip Select (CS), ativa a comunicação com um

slave, sendo que os outros 3 transmitem a informação: o master output – slave input (MOSI), o master input – slave output (MISO) e o clock série (SCK) [62].

O master fornece o sinal de clock que os slaves recebem e a velocidade de clock definida pelo master tem que ser uma que os slaves suportem. Todos os slaves que não foram ativados pelo CS ignoram o que o master envia pelo SCK e MOSI, não ativando o MISO. A simplicidade deste protocolo está representada na Figura 4.12, onde se faz o envio de 3 bytes de 8 bits. Ao contrário do I²C, não há overhead na transmissão de informação, pois não há a necessidade de enviar o endereço do slave nem bits extra de ACK, start e stop.

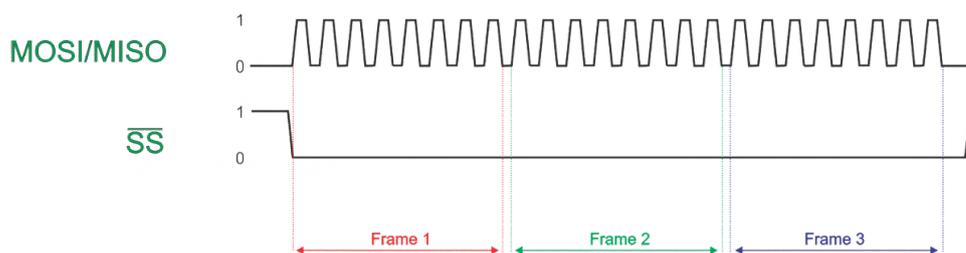


Figura 4.12: Exemplo de comunicação no protocolo SPI [9]

As vantagens deste protocolo são a rapidez com que se transmite dados e uma ligação rápida e simples entre um par master e slave. No entanto, este protocolo não tem, por defeito, confirmação de receção e envio de dados, requer vários fios para comunicar e não é muito aconselhável em situações com vários slaves, devido ao fio extra necessário por slave.

4.5 Discussão

Apresenta-se na Figura 4.13 o esquema de ligações. Como há aparelhos que funcionam a 5v e outros a 3,3v e como o master tem de comunicar com aparelhos que funcionam a diferentes tensões, é necessário realizar um shift de tensão bi-direcional .

Tendo em conta a comunicação necessária a realizar entre o processador e os dispositivos de medição (multi-slave), e com outro sistema (multi-master) decidiu-se utilizar o I²C. O facto de necessitar de poucos fios, de ocupar poucos pinos e ser um protocolo que implementa um sistema de *acknowledge* de transferência de informação tornam este protocolo o mais interessante.

Apesar da frame de I²C estar bem documentada, os dispositivos de medição podem demorar mais ou menos tempo a enviar a mesma quantidade de informação, como se demonstra na Figura 4.14, o que pode ter impacto na execução do algoritmo se os requerimentos temporais forem exigentes. Na comunicação com o IMU, verifica-se com facilidade como acontece a comunicação: o master envia 2 bytes, cada um com 9 bits pela linha SDA (representada a amarelo), seguindo-se o envio, por parte do IMU slave e a receção por parte do master, dos bytes de informação até o master der ordem para parar. A transmissão de bytes é mais fácil de visualizar a partir do sinal de clock SCL, a verde, que é sempre produzido pelo master. O clock está ativo quando se envia um byte, correspondendo a 9 bits, sendo possível verificar os tempos mortos entre os bytes, com

1 bit LOW no SCL entre bytes. Estes detalhes não estão explícitos na especificação do I²C pois depende da implementação realizada pelos instrumentos. No caso do GPS, a comunicação acontece de forma um bocado diferente: é necessário enviar um comando, e só depois mandar o byte corresponde ao registo que queremos ler. Os dois primeiros bytes correspondem ao envio do endereço do slave e do comando, o terceiro corresponde ao endereço do GPS que queremos aceder, e os bytes seguintes são o envio da informação por parte do GPS. Nesta imagem, é possível verificar um atraso maior entre bytes, que se deve à implementação do I²C por parte do GPS. Estes atrasos, que não estão especificados, podem ter impacto na duração da execução do algoritmo, sendo necessário medi-los para obter um valor concreto.

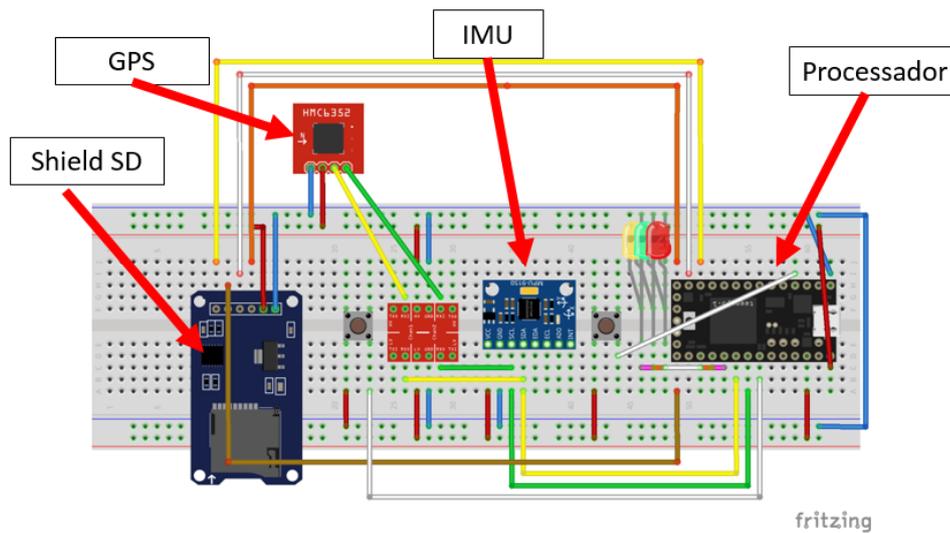


Figura 4.13: Esquema da montagem do hardware utilizado

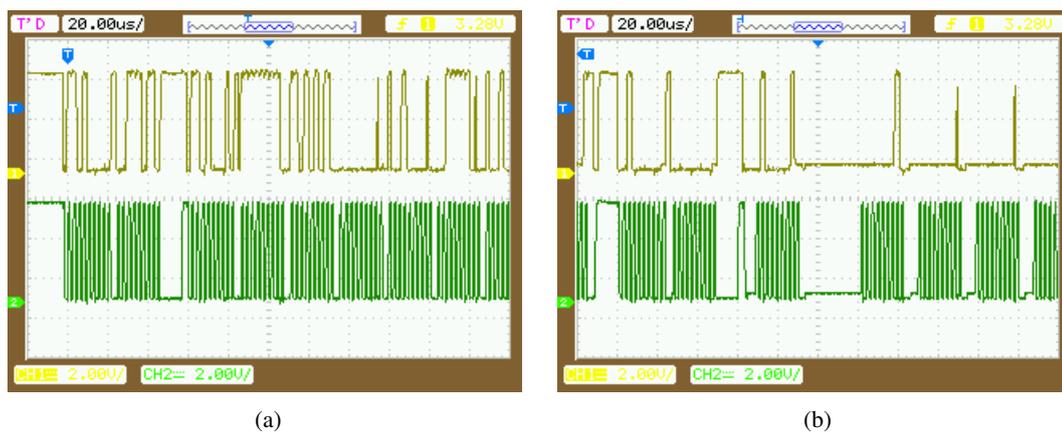


Figura 4.14: Medição de osciloscópio da comunicação por I²C do master com: (a) IMU; (b) GPS

Para enviar ou guardar informação recebida de um outro aparelho por I²C, é necessário configurar e definir os endereços de cada variável, que os outros aparelhos terão que conhecer para

conseguirem aceder.

Para concretizar o datalog, usar-se-á o protocolo SPI e o datalog será o único slave do processador neste protocolo.

Antes de se executar o algoritmo, é necessário verificar se os dispositivos de medição e se o cartão SD são detetados, procedendo-se à configuração dos mesmos e dos protocolos a utilizar. Se algum dispositivo não for detetado, o algoritmo não pode ser executado, e se não for detetado nenhum cartão SD, não será permitido realizar logs durante a execução do algoritmo. O algoritmo necessita de receber o vetor controlo do sistema principal, por isso só será executado após receber ordem do sistema central.

Os passos do ciclo de execução do algoritmo são: receber a informação necessária do sistema, ler os valores medidos e aplicar o filtro, armazenando os estados estimados nos endereços de memória respetivos. Por fim, receberá um pedido do sistema principal para enviar os estados estimados. Este ciclo repete-se a $100Hz$, executando o EKF_{TOTAL} em vez do EKF_{IMU} a cada $10Hz$, que é quando há uma nova leitura do GPS.

A implementação utiliza três leds para informar o utilizador acerca do estado do sistema. Através dos leds indica-se se o algoritmo está em funcionamento ou não e, antes da execução do algoritmo, indica-se se algum componente e dispositivo de medição não foi detetado. Colocou-se um botão para dar ordem de início ao sistema, ficando à espera da comunicação do sistema central, e outro botão para realizar um log dos dados lidos e estimados.

Capítulo 5

Resultados

5.1 Resultados de Simulação

Por forma a explorar o desempenho dos algoritmos propostos no Capítulo 3, foram desenvolvidas algumas manobras de teste através da ferramenta de simulação CarSim. Os resultados obtidos deste modelo de alta fidelidade foram utilizados como base para validar o algoritmo desenvolvido através de um script Matlab onde este foi implementado. Obteve-se sinais de testes para o vetor controlo e medições dos sensores, essenciais para o algoritmo.

Os algoritmos propostos foram testados em duas manobras típicas: a Double Line Change (DLC) e a Jturn. A primeira consiste numa alteração de faixa voltando para a inicial de seguida (esta manobra é típica numa situação de ultrapassagem ou desvio de obstáculos), enquanto que a Jturn consiste numa viragem constante, com o veículo a realizar uma curva. O *steering*, que é o conjunto de componentes e ligações que permitem virar um veículo com um certo ângulo, das duas manobras está demonstrado na Figura 5.1:

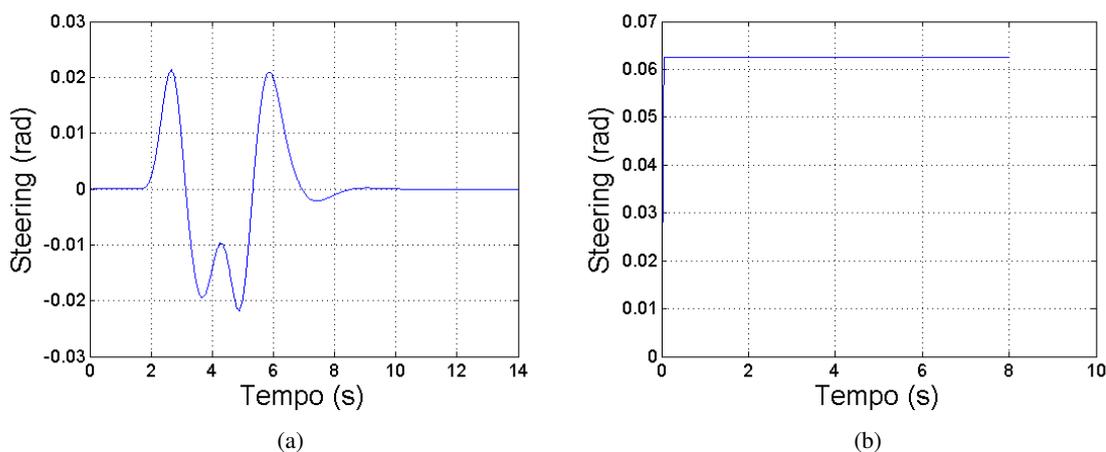


Figura 5.1: *Steering* das manobras: (a) DLC; (b) Jturn

Como os valores que entram no filtro são obtidos a partir do input do condutor na simulação,

a dinâmica da manobra poderá ter pequenas diferenças com a realização de uma nova simulação a diferentes velocidades.

Submeteram-se as duas versões do algoritmo aos sinais de teste. Os filtros de cada algoritmo foram afinados e configurados para a manobra DLC a 90km/h , de forma a obter os melhores resultados possíveis para essa manobra em específico. Para se demonstrar a versatilidade do algoritmo, vai-se submeter a mesma configuração às outras velocidades e, inclusive, a outra manobra, na secção posterior.

Adicionou-se ruído branco às medições, com os valores na Tabela 5.1:

Tabela 5.1: Ruído das medições

	Ruído
Aceleração	0,025 (m/s^2)
Velocidade	0,10 (m/s)
Velocidade angular	0,11 ($^\circ/s$)
Direção	2,5 ($^\circ$)
Slip angle	1,0 ($^\circ$)
Posição	0,10 (m)

Os valores de aceleração e da velocidade angular são semelhantes aos valores do desvio padrão as medições do IMU MPU-9250. Para a direção, considerou-se o ruído presente no cálculo do ângulo a partir da função atan2 , considerando-se a medição dos eixos y e x do magnetómetro. Para definir o ruído da velocidade, considerou-se um valor arbitrário com uma magnitude de ordem superior à da aceleração.

Para o *slip angle*, que não é medido mas sim calculado, considera-se um ruído arbitrário mas considerado razoável. Assume-se que a direção do veículo usada para calcular o *side slip* é a estimada pelo algoritmo 2, subtraindo-se a direção do movimento dada pelo GPS. Uma vez que a precisão definida na *datasheet* é obtida com o recetor numa posição estática, considera-se o ruído para esta variável, numa situação dinâmica, duas vezes superior ao indicado. Como consequência da medição ser resultado de um cálculo, a confiança dada à mesma é muito reduzida, quer no EKF_{IMU} quer no EKF_{TOTAL} , sendo o modelo determinante para estimar corretamente este ângulo.

Para a posição, considerou-se um ruído reduzido como foi analisado na Secção 4.3, considerando-se o indicado pela *datasheet*. No entanto, a precisão do GPS é variável e pode apresentar um erro significativo ($> 2m$) sendo que, para fins de simulação, a medição de posição obtida do modelo de simulação tem uma precisão elevada.

A ruído branco, de média zero, está definido como v_k^{A1} para o algoritmo proposto 1 e v_k^{A2} para o algoritmo proposto 2:

$$v_k^{A1} = \begin{cases} 0,25 \\ 0,025 \\ 0,11\pi/180 \end{cases}$$

$$v_k^{A_2} = \begin{cases} 0,1 \\ 0,1 \\ 0,025 \\ 0,025 \\ 2,5\pi/180 \\ 0,11\pi/180 \\ 1,0\pi/180 \\ 0,1 \\ 0,1 \end{cases}$$

O ruído de processo w_k é, para fins de simulação, considerado zero para os dois algoritmos.

A afinação do filtro passa por variar as matrizes \mathbf{Q} e \mathbf{R} , como já foi discutido. O ponto de partida da matriz \mathbf{R} será o valor do ruído, definido na matriz v_k , e \mathbf{Q} começará por ser definida com valores pequenos, aumentando-se gradualmente cada valor até se conseguir uma resposta satisfatória. \mathbf{R} poderá sofrer alterações, uma vez que influencia o desempenho da matriz \mathbf{Q} . Tendo-se em conta que um valor mais pequeno de \mathbf{Q} resulta numa maior confiança no cálculo resultante do modelo e um valor alto resulta num maior peso dado à medição, considerou-se as matrizes $\mathbf{Q}_{IMU}^{A_1}$, $\mathbf{Q}_{TOTAL}^{A_1}$, $\mathbf{R}_{IMU}^{A_1}$ e $\mathbf{R}_{TOTAL}^{A_1}$ para o algoritmo 1, e $\mathbf{Q}_{IMU}^{A_2}$, $\mathbf{Q}_{TOTAL}^{A_2}$, $\mathbf{R}_{IMU}^{A_2}$ e $\mathbf{R}_{TOTAL}^{A_2}$ para o algoritmo 2:

$$\mathbf{R}_{IMU}^{A_1} = \begin{bmatrix} 0,000001 & 0 & 0 \\ 0 & 0,025 & 0 \\ 0 & 0 & 1,5\pi/180 \end{bmatrix}$$

$$\mathbf{R}_{TOTAL}^{A_1} = \begin{bmatrix} 0,01 & 0 & 0 \\ 0 & 0,025 & 0 \\ 0 & 0 & 1,5\pi/180 \end{bmatrix}$$

$$\mathbf{Q}_{IMU}^{A_1} = \begin{bmatrix} 0,01 & 0 & 0 & 0 & 0 \\ 0 & 0,5 & 0 & 0 & 0 \\ 0 & 0 & 0,002 & 0 & 0 \\ 0 & 0 & 0 & 1000 & 0 \\ 0 & 0 & 0 & 0 & 1000 \end{bmatrix}$$

$$Q_{TOTAL}^{A_1} = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 \\ 0 & 0,5 & 0 & 0 & 0 \\ 0 & 0 & 0,002 & 0 & 0 \\ 0 & 0 & 0 & 1000 & 0 \\ 0 & 0 & 0 & 0 & 1000 \end{bmatrix}$$

$$R_{IMU}^{A_2} = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0,025 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0,025 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100\pi/180 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1,5\pi/180 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1000 \end{bmatrix}$$

$$R_{TOTAL}^{A_2} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0,1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0,025 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0,025 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0,5\pi/180 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1,5\pi/180 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0,001 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Q_{IMU}^{A_2} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0,01 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0,006 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0,001 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0,1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0,1 \end{bmatrix}$$

$$Q_{TOTAL}^{A_2} = \begin{bmatrix} 0,1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0,001 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0,006 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0,001 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0,01 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0,01 \end{bmatrix}$$

Os valores foram definidos de forma a dar mais peso (**Q** pequeno) ao modelo, ou mais peso (**Q** elevado) às medições. Quando uma medida não está disponível, dá-se um valor mais pequeno, apesar da variação deste valor para medidas que não estão definidas no modelo das medições não ter um impacto relevante no desempenho do filtro. Há algumas exceções, pois o filtro obteve um desempenho superior com esses valores. Alguns valores de **R** também foram alterados em relação aos valores iniciais, v_k , sendo estas as matrizes que funcionam melhor com a respetiva matriz **Q**.

A equação de *drag*, componente inserida no algoritmo 2, é calculada com os valores $\rho = 1,225$, $A = 0,8l_s$ e $C = 0,32$:

$$F_{DRAG} = \frac{1}{2} \rho C_D A v^2$$

Considerou-se uma pressão atmosférica a 15°C e ao nível do mar e uma área frontal de acordo com o veículo da simulação. O coeficiente de *drag* está, também, associado à aerodinâmica do veículo, tendo-se considerado um valor de acordo com o veículo da simulação.

5.1.1 Double Line Change: DLC

Foram consideradas as velocidades 60, 90, 120 e 160 km/h para se submeter os algoritmos a níveis de exigência diferentes, dentro do mesmo tipo de movimento. Uma manobra mais rápida implica submeter o veículo a forças mais elevadas, dificultando a estimação das variáveis de estado. Testar os algoritmos com diferentes velocidades permite desenvolver um filtro que seja consistente e que apresenta um bom desempenho para um maior leque de manobras a diferentes velocidades.

Na Figura 5.2 apresenta-se o resultado da estimação dos dois algoritmos, comparativamente com o sinal de teste afetado pelo ruído. Todos os resultados serão apresentados desta forma.

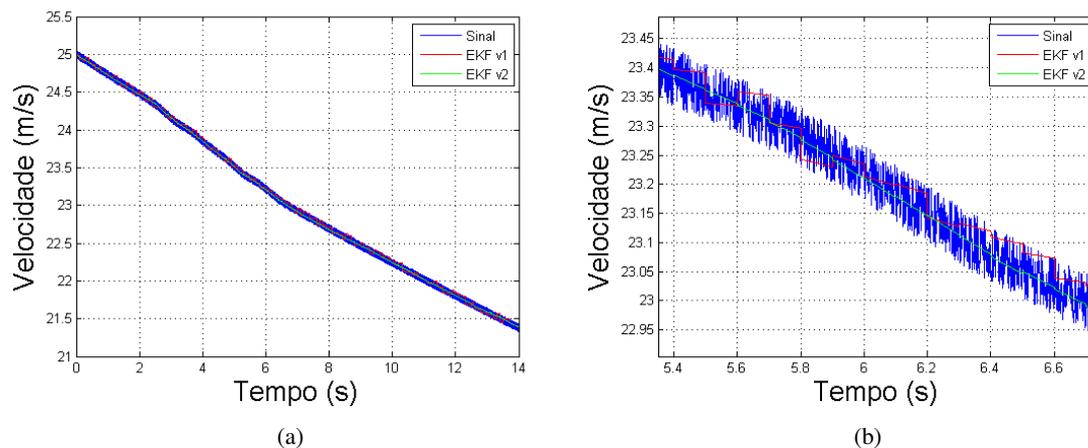


Figura 5.2: Velocidade Longitudinal da manobra DLC a 90km/h: (a) Gráfico da estimação; (b) Ampliação do gráfico (a)

Observa-se uma melhoria considerável no algoritmo 2. A estimação acompanha bastante melhor o sinal verdadeiro, devendo-se principalmente à introdução da componente de *drag*. Como a estimação é mais precisa e o erro não aumenta durante a aplicação do EKF_{IMU} , o algoritmo, com a aplicação do EKF_{TOTAL} não aplica uma correção tão significativa como acontece no algoritmo 1. No algoritmo 1, é possível verificar os momentos (de frequência $10Hz$) em que o EKF_{TOTAL} efetua uma correção, corrigindo a estimação que fica gradualmente pior. Este resultado também demonstra o desempenho reduzido do EKF quando o modelo não linear não é fiel ao que acontece na realidade, dificultando o processo de aprendizagem e correção do filtro.

A velocidade lateral é apresentada na Figura 5.3, onde é possível observar um desempenho muito semelhante entre os dois algoritmos:

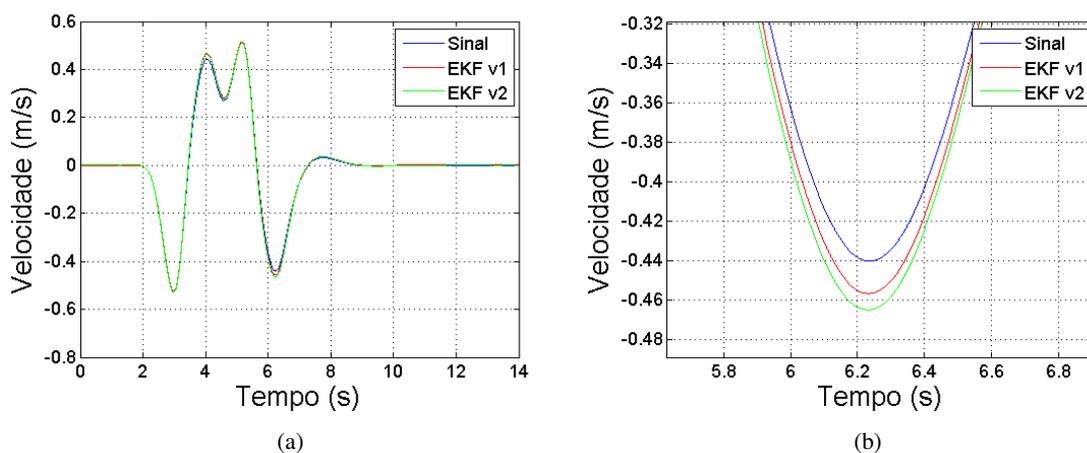


Figura 5.3: Velocidade Lateral da manobra DLC a 90km/h: (a) Gráfico da estimação; (b) Ampliação do gráfico (a)

Na Figura 5.4, apresenta-se o erro normalizado da estimação:

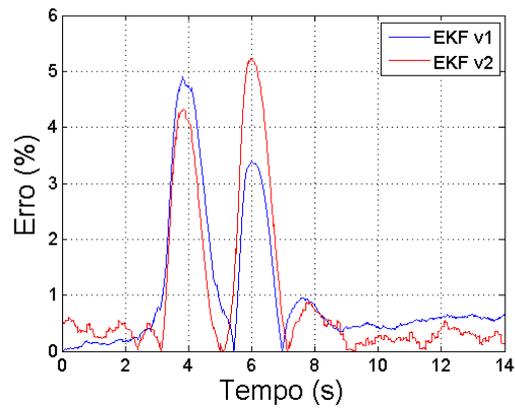


Figura 5.4: Erro de estimação da Velocidade Lateral da manobra DLC a 90km/h dos dois algoritmos

Ambas as estimções têm um desempenho muito semelhante, havendo uma variação no tempo em que conseguem o melhor resultado. Esta variação é uma consequência da afinação definida para cada algoritmo mas, como se esperava, o resultado não melhorou significativamente com o aumento do estado no algoritmo 2.

A velocidade angular tem um desempenho muito semelhante nas duas versões, apresentado na Figura 5.5:

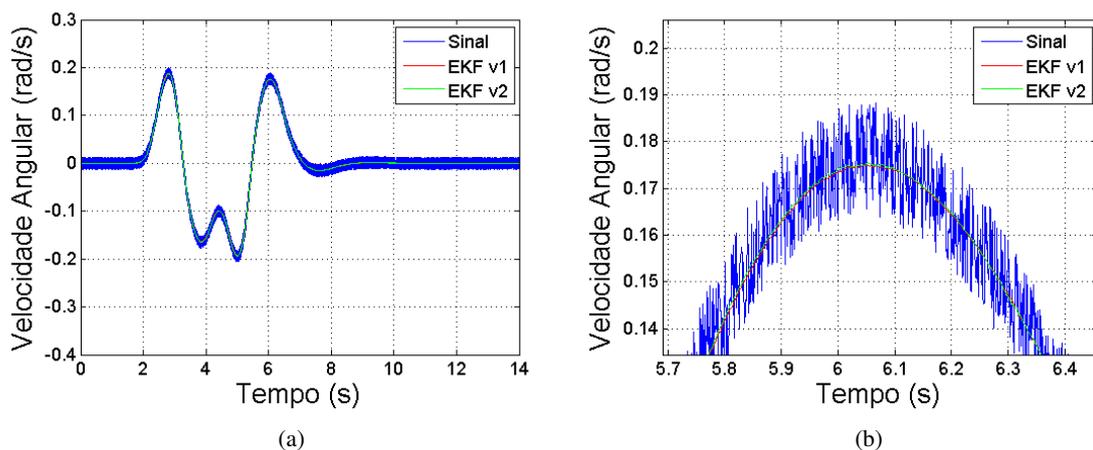


Figura 5.5: Velocidade Angular da manobra DLC a 90km/h: (a) Gráfico da estimação; (b) Ampliação do gráfico (a)

Apresenta-se também o erro na Figura 5.6:

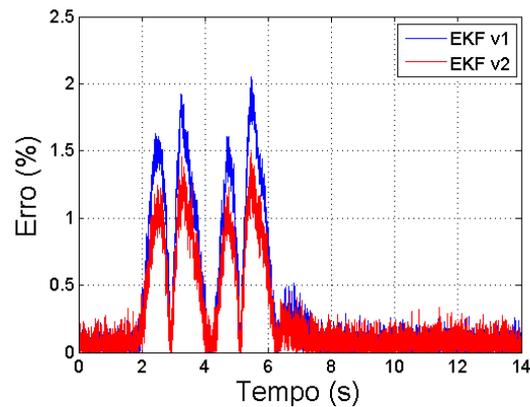


Figura 5.6: Erro de estimação da Velocidade Angular da manobra DLC a 90km/h dos dois algoritmos

Os resultados apresentados na Figura 5.5 e Figura 5.6 demonstram uma ligeira melhoria do segundo algoritmo face ao primeiro.

A aceleração lateral é definida a partir do modelo das medições e está representada na Figura 5.7:

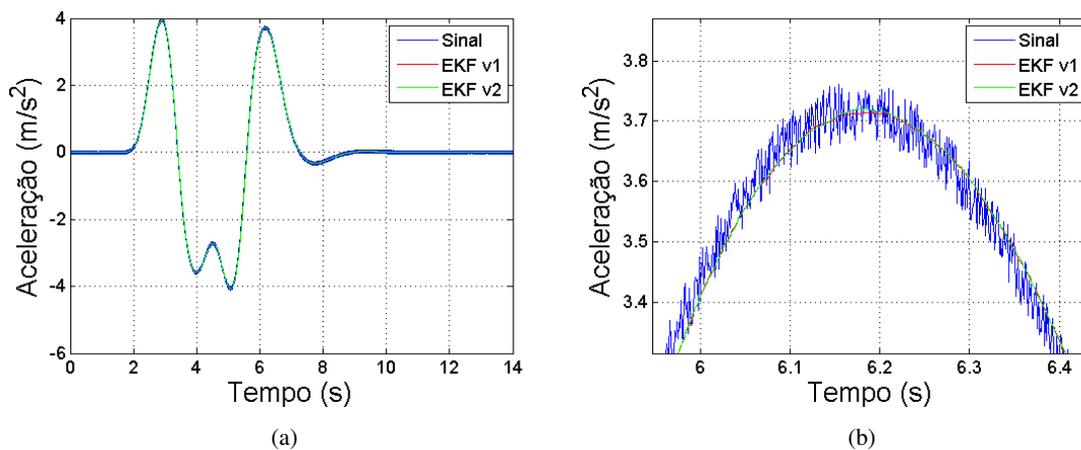


Figura 5.7: Aceleração Lateral da manobra DLC a 90km/h: (a) Gráfico da estimação; (b) Ampliação do gráfico (a)

O resultado é praticamente idêntico nos dois algoritmos, o que é esperado visto que o modelo das medições é idêntico. Com base neste resultado, pode-se afirmar que a diferença que se verifica na velocidade lateral do algoritmo 2 deve-se à estimação de outras variáveis que dependem da velocidade lateral, da sua medição e de uma afinação diferente.

As forças laterais têm uma estimação, também, idêntica. Na Figura 5.8 apresenta-se a força lateral dianteira e traseira, na Figura 5.9 uma ampliação da força lateral dianteira e na Figura 5.10 o erro da força lateral dianteira:

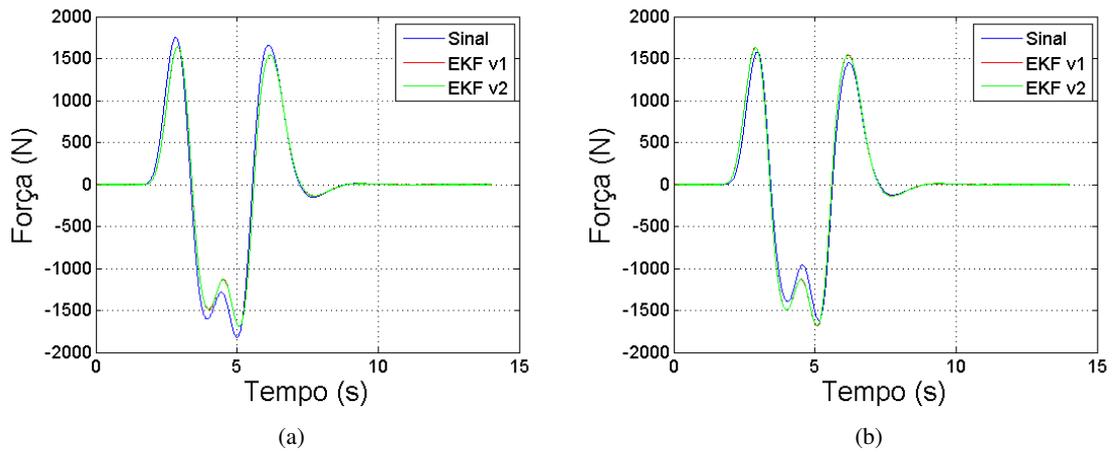


Figura 5.8: Força Lateral da manobra DLC a 90km/h: (a) Força Lateral Dianteira; (b) Força Lateral Traseira

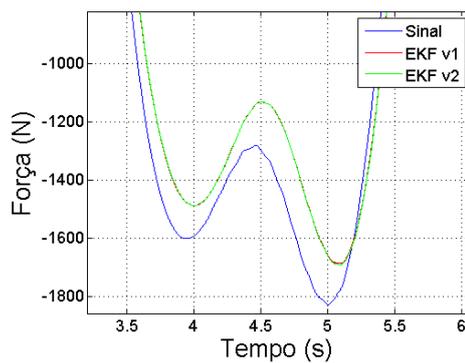


Figura 5.9: Ampliação da estimação da Força Lateral Dianteira da manobra DLC a 90 km/h dos dois algoritmos

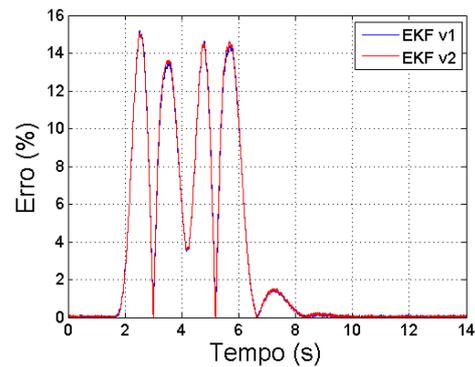


Figura 5.10: Erro da estimação da Força Lateral Dianteira da manobra DLC a 90 km/h dos dois algoritmos

Estas forças são estimadas de forma idêntica pelos dois algoritmos, como é possível observar pelo erro representado na Figura 5.10, que é igual para os dois algoritmos. Não é apresentado o erro da força lateral traseira pois tem um comportamento idêntico.

O algoritmo 1 não estima mais informação sobre o sistema, sendo este o máximo de informação que se consegue obter. O algoritmo 2 é uma evolução nesse aspeto, estimando um maior conjunto de informação. A direção, o *slip angle* e a posição do sistema são informação relevante e pode ser interessante ter acesso a esses dados.

A direção resulta da integração da velocidade angular durante o EKF_{IMU}^{A2} , sendo corrigida pelo EKF_{TOTAL}^{A2} . O resultado é apresentado na Figura 5.11:

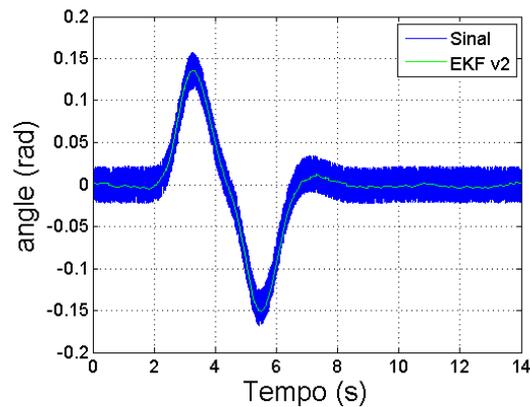


Figura 5.11: Estimação da Direção do Veículo da manobra DLC a 90km/h

É possível observar uma estimação muito boa desta variável, eliminando quase totalmente o efeito do ruído da medição. Sem filtrar o sinal vindo do dispositivo de medição, não será possível usar a direção que o mesmo mede, devido ao ruído. A introdução desta informação no estado do algoritmo apresenta-se como uma boa solução para obter uma estimação sem ruído.

Este algoritmo também tem um desempenho muito bom na estimativa do *slip angle*, apresentado na Figura 5.12:

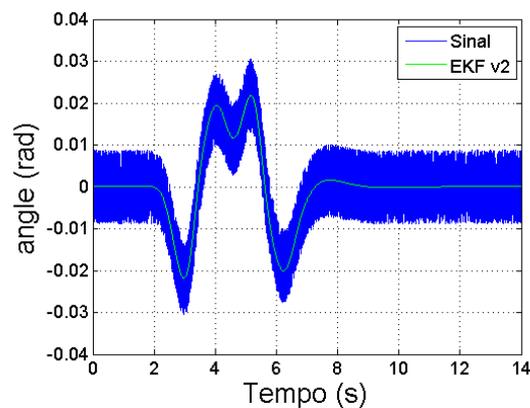


Figura 5.12: Estimação do Slip Angle do Veículo da manobra DLC a 90km/h

Uma vez que se dá pouca confiança à medição do *slip angle*, o resultado desta estimativa é quase exclusivamente resultado do modelo, obtendo-se uma boa estimativa, sem ser necessário recorrer à medição para a corrigir.

A posição também obtém resultados satisfatórios, apresentados na Figura 5.13. Uma vez que se obtém uma boa estimativa da velocidade, a posição que resulta da sua integração também será boa, usando a ocasional correção para corrigir certos *drifts*. No entanto, é necessário ter em conta que as medições aqui utilizadas têm uma precisão elevada, o que não se verifica nas medições realizadas por um recetor GPS. Na prática, pode haver uma discrepância grande entre a posição

obtida pelo GPS e o que é calculado pelo modelo, o que irá afetar o processo de aprendizagem e correção do algoritmo, piorando a estimação dos outros estados.

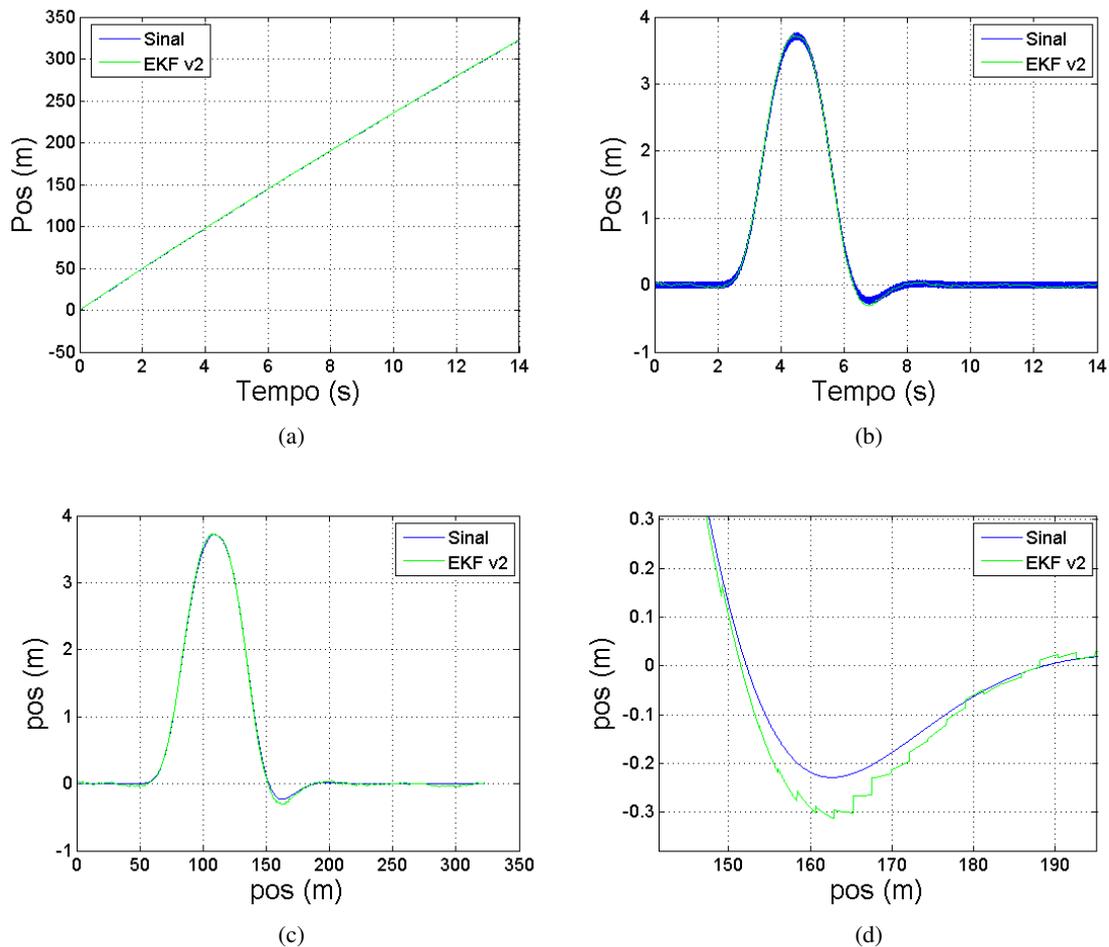


Figura 5.13: Estimação da posição da manobra DLC a 90km/h: (a) Posição em x; (b) Posição em y; (c) Posição x vs y; (d) Ampliação do gráfico (c)

Estes resultados foram obtidos com os sinais de teste a 90km/h. Ambos os algoritmos têm um desempenho e comportamento semelhante para todas as velocidades, conseguindo a convergência e uma boa estimação dos estados.

Os algoritmos obtidos para a manobra DLC foram testados a diferentes velocidades e, devido à semelhança dos resultados obtidos, apenas são apresentados os resultados completos para a manobra a 90Km/h. O erro da velocidade lateral é analisado para diferentes velocidades, uma vez que esta variável apresenta o maior grau de incerteza, sendo os resultados apresentados na Figura 5.14:

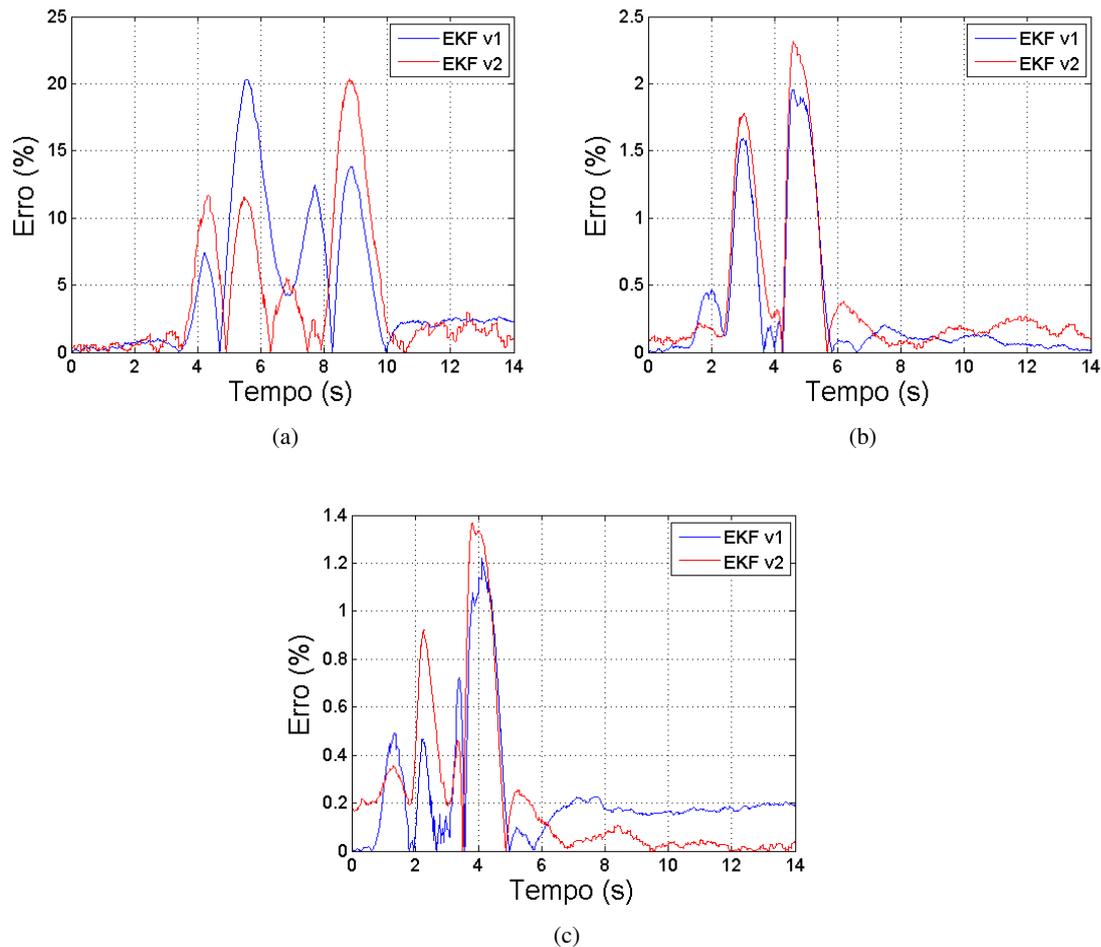


Figura 5.14: Erro da Estimação da Velocidade Lateral dos dois algoritmos para a manobra DLC: (a) a 60km/h; (b) a 120km/h; (c) a 160km/h

Observa-se que os dois algoritmos têm um desempenho semelhante a todas as velocidades apesar de se ter usado uma configuração igual para todas, definida para a manobra a 90km/h. Observa-se, também, que a percentagem de erro diminui com o aumento da velocidade. Isto deve-se à magnitude dos valores que são estimados, pois a uma velocidade baixa, a velocidade lateral que se verifica é muito baixa ($< 0,5m/s$), sendo a estimacão desta variável afetada pelo erro das outras estimacões. A uma velocidade mais elevada, a velocidade lateral é mais elevada, sendo a sua estimacão menos afetada pelo ruído das outras estimacões, que não aumenta significativamente com a velocidade.

5.1.2 Jturn

O *steering* constante da manobra Jturn simulada provoca uma dinâmica lateral bastante mais pronunciada do que na manobra DLC, elevando a dificuldade da sua estimacão. Será utilizada a mesma configuracão apresentada para a manobra DLC, sendo esta simulacão importante para avaliar a versatilidade dos dois algoritmos e como se adaptam a uma situacão completamente

diferente. Por uma questão de simplicidade e clareza dos resultados, apenas serão apresentados os resultados da estimação da velocidade lateral e os erros associados à mesma uma vez que é esta a variável de estado que apresenta uma maior dificuldade de estimação. Apresenta-se, também, o erro da velocidade angular, demonstrando as semelhanças com o respetivo erro da DLC.

Consideraram-se 3 manobras com as velocidades de 60, 90 e 120 km/h e com rotação de 60°.

Na Figura 5.15 apresenta-se o desempenho dos dois algoritmos na estimação da velocidade lateral:

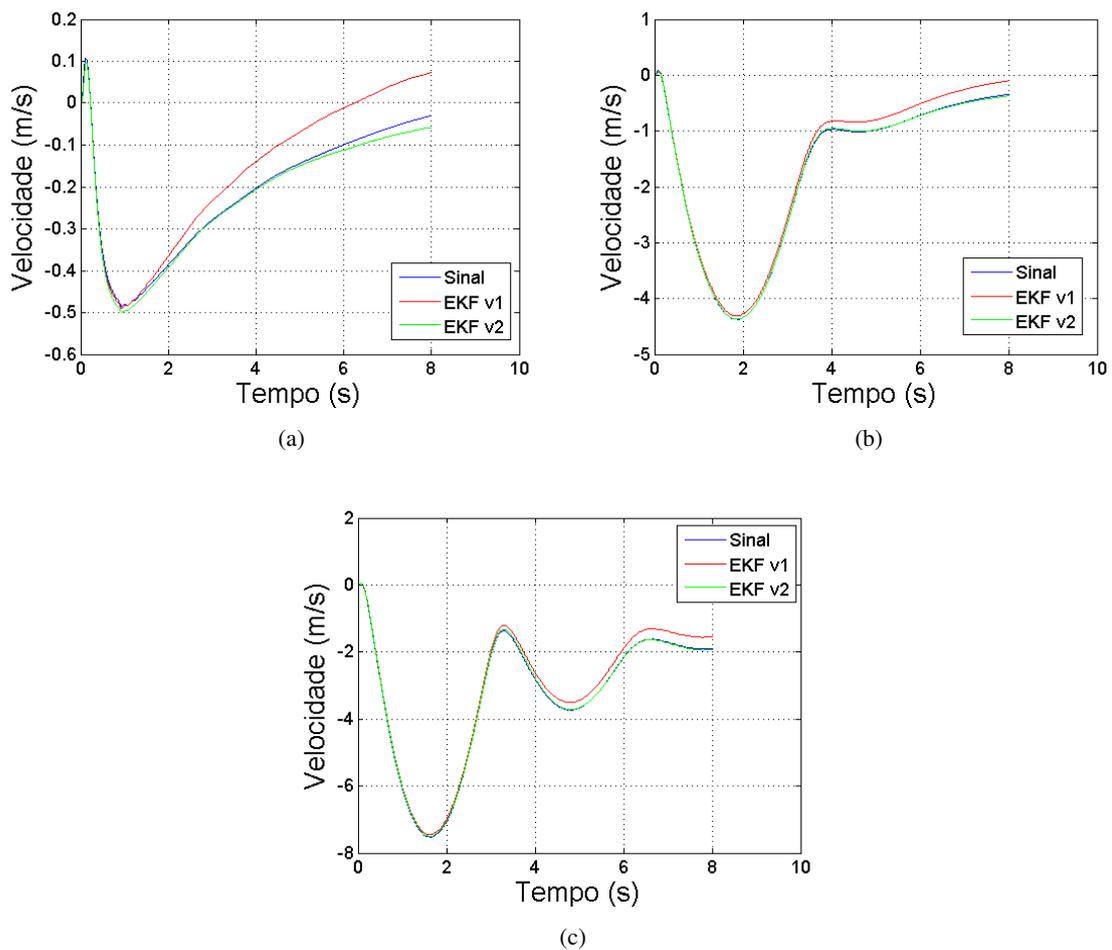


Figura 5.15: Estimação da Velocidade Lateral dos dois algoritmos para a manobra Jturn: (a) a 60km/h; (b) a 90km/h; (c) a 120km/h

observa-se que o algoritmo 1 não consegue acompanhar totalmente o sinal de teste a nenhuma velocidade, enquanto que o algoritmo 2 está mais perto, sugerindo que enquanto houver rotação do veículo, o algoritmo não conseguirá convergir. Analisando o erro, na Figura 5.16:

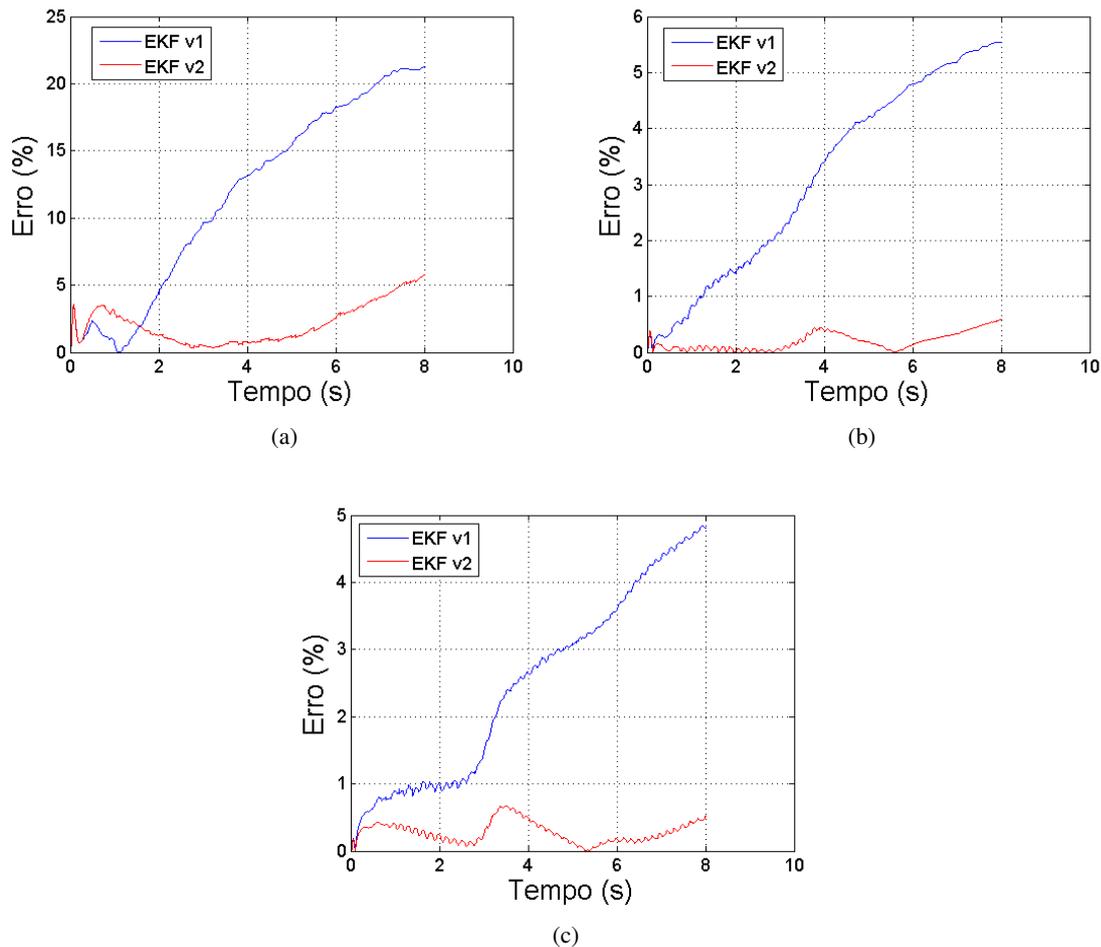


Figura 5.16: Erro da Estimação da Velocidade Lateral dos dois algoritmos para a manobra Jturn: (a) a 60km/h; (b) a 90km/h; (c) a 120km/h

Confirma-se que o erro da estimação do algoritmo 1 cresce durante a duração da manobra. O algoritmo 2 consegue reduzir o erro para valores aceitáveis, mas não conseguindo a convergência total.

O erro da velocidade angular é apresentado na Figura 5.17. Observa-se que o erro do algoritmo 1 é, constantemente, superior ao do algoritmo 2, sendo o último capaz de reduzir o erro para valores quase nulos. Este erro do algoritmo 1 é a causa da não convergência do algoritmo 1 na estimação da velocidade lateral. Enquanto que na DLC esse atraso não tem grande impacto, numa manobra com uma dinâmica lateral maior como a Jturn do sinal de teste esse erro irá ser integrado pelo modelo no cálculo da velocidade lateral, resultando num pequeno *drift* que irá durar enquanto a rotação do veículo não acabar. O facto do algoritmo 2 recorrer à medição da velocidade lateral para corrigir erros de estimação, juntamente com reduzido erro da velocidade angular, resultam numa estimação de maior qualidade.

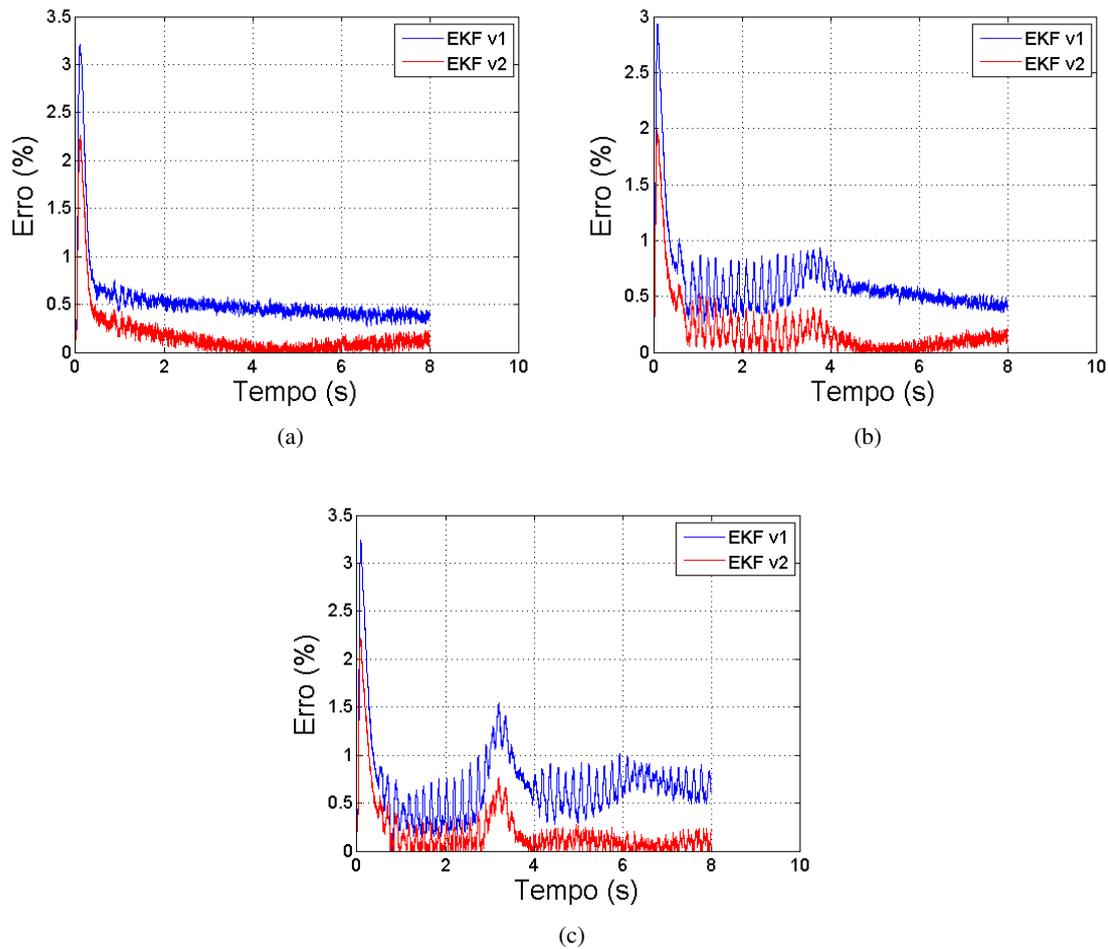


Figura 5.17: Erro da Estimação da Velocidade Angular dos dois algoritmos para a manobra Jturn: (a) a 60km/h; (b) a 90km/h; (c) a 120km/h

5.1.3 Discussão

Apesar do algoritmo 2 ser uma iteração mais recente do algoritmo 1, ambos são opções válidas para aplicar na prática. O algoritmo 1 é um algoritmo menos pesado computacionalmente, devido ao tamanho reduzido das matrizes do filtro, não tendo um desempenho significativamente inferior nas variáveis de estado que estima. No entanto, em manobras com uma dinâmica lateral mais pronunciada, este algoritmo tem um desempenho inferior, não convergindo enquanto a manobra não terminar.

O algoritmo 1 apresenta uma boa estimativa das variáveis de estado para a manobra DLC. O algoritmo 2 consegue uma melhoria em algumas estimativas, nomeadamente na da velocidade longitudinal com uma melhoria significativa, mas obtém uma estimativa igual ou ligeiramente pior da velocidade lateral. Tendo em conta a quantidade de informação a mais que o algoritmo 2 processa e estima comparado com o algoritmo 1, considera-se que ambos os algoritmos desenvolvidos são aceitáveis e aplicáveis na prática para esta manobra.

O algoritmo 2 estima um maior conjunto de informação e tem acesso a mais medições, obtendo resultados interessantes em manobras com uma dinâmica lateral mais pronunciada. Neste algoritmo entra a medição da velocidade lateral, o que pode não ser muito útil numa manobra como a DLC uma vez que não há uma melhoria em relação à estimação do algoritmo 1.

O comportamento dos algoritmos pode ser alterado variando, principalmente, a matriz \mathbf{Q} , e o comportamento do EKF depende bastante da afinação. Na matriz \mathbf{Q} atribuiu-se às forças laterais, que são desconhecidas, um valor muito alto, indicando que o filtro deve seguir-se pelas medições e estimações de outras variáveis para estimar estas variáveis. Quando não temos muita confiança na medição de uma variável, como é o caso do *slip angle*, o valor é reduzido, favorecendo-se o modelo. Esta modelização permite a aplicação do algoritmo, projetado para uma manobra, em outras manobras, caso contrário teria-se que alterar esta matriz, juntamente com o modelo, para acomodar manobras e pavimentos diferentes.

Nesta secção apresentaram-se os resultados para as duas manobras usando a mesma afinação, que foi obtida para a manobra DLC a 90km/h. Seria possível obter melhores resultados com o algoritmo 1 para a manobra Jturn se o algoritmo fosse afinado para essa manobra, mas nunca seria possível obter a convergência total. O algoritmo 2 revela-se como um algoritmo mais versátil, onde uma afinação realizada para uma manobra obteve resultados muito bons para manobras diferente e de maior dificuldade.

Capítulo 6

Conclusão e Trabalho Futuro

6.1 Conclusão

Nesta dissertação propôs-se o desenvolvimento de um algoritmo de fusão sensorial para um sistema de posicionamento tendo sido aplicado a um caso de estudo de um veículo terrestre, de forma a combinar informação de dispositivos de medição diferentes, com o objetivo de melhorar a informação obtida pelos mesmos e ultrapassar as suas limitações individuais. A fusão entre dispositivos permite obter um conjunto de informação mais interessante e de melhor qualidade, que pode ser utilizado por variados veículos ou sistemas autónomos para se posicionarem com precisão.

Foi necessário identificar os requisitos do sistema e identificar dispositivos de medição, a utilizar, que permitissem cumprir esses requisitos, tendo em conta um orçamento reduzido. Estabeleceu-se a comunicação com os dispositivos de medição e definir parâmetros de configuração dos mesmos que alteram o sinal obtido. Foi necessário realizar o processamento dos sinais obtidos do dispositivo de medição, que inclui a conversão do output dos mesmos para unidades SI através de expressões definidas pelos fabricantes dos dispositivos. Diferentes tipos de dispositivos de medição, ou mesmo dispositivos do mesmo tipo mas de fabricantes diferentes, têm implementações diferentes que podem levar a um tratamento de informação diferente. Verificou-se, por exemplo, no magnetómetro em relação ao acelerómetro e ao giroscópio, que fazem parte do mesmo IMU mas os passos necessários para o utilizar são diferentes.

De forma a integrar os diferentes componentes do sistema, foi necessário definir um protocolo de comunicação. Nesta decisão considerou-se a dificuldade de implementação, a escalabilidade do protocolo para ligar mais componentes se necessário, a velocidade da transferência de informação e a capacidade de se ligar vários masters e slaves. Identificou-se protocolos que seriam capazes de estabelecer a comunicação que respeita-se as condições definidas, escolhendo-se um muito usado em sistemas de engenharia e que muitos dispositivos de medição já estão preparados para utilizar.

Desenvolveram-se dois algoritmos de fusão sensorial, baseados em Extended Kalman Filter, que estimam informação relativa ao posicionamento de um veículo terrestre com recurso a medições de dois dispositivos de medição, um IMU e um recetor GPS. Além de combinarem infor-

mação de naturezas diferentes, os algoritmos conseguem estimar variáveis que não são medidas e melhorar as que são, filtrando o ruído. Algumas ferramentas, como o Matlab, foram fundamentais para se obter um algoritmo que produzi-se resultados aceitáveis. Ficou clara a importância de identificar software que reúna as condições ideais, como a facilidade na alteração de parâmetros do algoritmo e a capacidade de realizar testes.

Os algoritmos foram submetidos a testes de simulação, conseguindo estimar, com precisão, informação como a velocidade e posição do veículo, que são fundamentais num sistema de posicionamento, com recurso a um modelo que descreve o movimento de um veículo e a medições do IMU e do GPS. O objetivo proposto foi atingido, obtendo-se uma estimativa com qualidade dessa informação.

6.2 Trabalho Futuro

O trabalho futuro consistirá na aplicação prática do algoritmo desenvolvido, com a realização de testes para avaliar o desempenho do algoritmo num ambiente menos controlado, e na caracterização do erro do recetor GPS em movimento, teste para o qual não se conseguiu reunir condições para realizar, nem foi definido um método para o realizar. Os algoritmos também podem ser alvo de melhorias, principalmente se se considerar a possibilidade de usar outro tipo de dispositivos de medição, aumentando o leque de informação disponível sobre o veículo e que os algoritmos poderiam usar para melhorar a estimação.

Uma limitação do algoritmo 2 é considerar-se a medição do *slip angle* (que não é realmente medido) em vez de se considerar as medições de direção provenientes do GPS e do IMU, o que permitiria obter um algoritmo mais robusto e que processaria diretamente as medições dos dispositivos de medição, mas é algo difícil de conseguir em ambiente de simulação, com sinais de teste. Os algoritmos foram testados para um conjunto de manobras específicas e de curta duração, não tendo sido testados para uma grande variedade situações reais. No entanto, a realização de testes práticos permitiria ultrapassar estas limitações e permitiria, também, perceber o comportamento dos dispositivos de medição em situações dinâmicas, nomeadamente em termos de ruído. Essa caracterização será importante na transição para uma aplicação prática, pois o desempenho do algoritmo poderá sofrer com essas diferenças.

Anexo A

Script para Comunicação Série com Teensy

Este script lê um ficheiro *.csv* com os vetores **U** e **Z** dos sinais de teste, e envia-os pela porta série para o micro-processador, recebendo e guardando num ficheiro *.txt* a resposta do mesmo.

A.1 Código Python

```
import csv #https://docs.python.org/2/library/csv.html
from serial import Serial # para comunicar pela porta serie
import struct # para criar inteiros de 4bits
import binascii # para fazer print dos valores HEX das variaveis

# inicializacao dos vectores
U1=[]
U2=[]
U3=[]
U4=[]
U5=[]
Z1=[]
Z2=[]
Z3=[]
Z4=[]

i = 0

# le o ficheiro com os vectores U e Z
print 'A ler ficheiro'
with open('measurements_to_csv_v2.csv', 'rb') as f: #abre o ficheiro csv
```

```
reader = csv.reader(f)

for row in reader: #le linha a linha
    #guarda a respectiva coluna
    U1.append((row[0]))
    U2.append((row[1]))
    U3.append((row[2]))
    U4.append((row[3]))
    U5.append((row[4]))
    Z1.append((row[5]))
    Z2.append((row[6]))
    Z3.append((row[7]))
    Z4.append((row[8]))
    i=i+1

# abre porta serie
print 'Opening serial port'
ser = Serial('COM5', 9600, timeout=3)
# da um tempo para o arduino inicializar
time.sleep(3)

# mandar bit de start para "sincronizar" com o arduino
start = 254
send = ser.write(struct.pack('B',start))
print "start " +binascii.hexlify(struct.pack('B',start))
ser.readline() # espera por resposta

s = ""
resp = ""
string = ''
num = -5000
writer = open('responses.txt','w')
print 'Comm begin'
for j in range(i):
    print j
    # envia os vectores
    send = ser.write(struct.pack('l',int(float(U1[j]))))
    send = ser.write(struct.pack('l',int(float(U2[j]))))
    send = ser.write(struct.pack('l',int(float(U3[j]))))
    send = ser.write(struct.pack('l',int(float(U4[j]))))
```

```
send = ser.write(struct.pack('l',int(float(U5[j]))))
send = ser.write(struct.pack('l',int(float(Z1[j]))))
send = ser.write(struct.pack('l',int(float(Z2[j]))))
send = ser.write(struct.pack('l',int(float(Z3[j]))))
send = ser.write(struct.pack('l',int(float(Z4[j]))))

# para receber matrizes
#print ser.readline();

# recebe-os

print "Tempo de execucao EKF em us: " + ser.readline()

print "recebeu:"
resp1 = ser.readline();
resp2 = ser.readline();
resp3 = ser.readline();
resp4 = ser.readline();
resp5 = ser.readline();
resp6 = ser.readline();
resp7 = ser.readline();

# prepara string para guardar no ficheiro txt
s = str(float(resp1)/100000)+","+str(float(resp2)/100000)+"," ...
    +str(float(resp3)/100000)+","+str(float(resp4)/100000)+"," ...
    +str(float(resp5)/100000)+","+str(float(resp6)/100000)+"," ...
    +str(float(resp7)/100000)
string = s.replace('\r\n','')
para ficar tudo na mesma linha
string = string+'\n'# no fim da um newline
#print binascii.hexlify(string)
print string
#writer.writerow(s)
```

```
writer.write(string) # guarda resposta no ficheiro

string = ''

ser.close()
```

Anexo B

Erro da Projeção Equirectangular

B.1 Código Matlab

```
R = 6371000;          % raio da Terra em metros

% ponto 1 é o ponto da origem, as distâncias são todas calculadas a este
% valores em graus
latitudes = [41.1785288,41.1784602,41.1856305,41.1781372,41.1693507, ...
            41.1781533,41.1791869,41.1786217,41.178670100000005,41.1429843, ...
            41.228249,41.1817548,41.1838219];
longitudes = [-8.5956001,-8.5882187,-8.5958576,-8.6044407,-8.5942268, ...
            -8.5955358,-8.595686,-8.596265300000002,-8.5947418,-8.5926819, ...
            -8.6016083,-8.6613464,-8.5350037];

for i=1:length(latitudes)

    % projeção
    x_2(i) = R * deg2rad(longitudes(i))*cos(deg2rad(latitudes(1)));
    y_2(i) = R * deg2rad(latitudes(i));

    distXY_2(i)= sqrt((x_2(i)-x_2(1))^2+(y_2(i)-y_2(1))^2);

    % calcular distâncias a partir da fórmula haversine
    difLat(i) = deg2rad(latitudes(i)-latitudes(1));
    difLon(i) = deg2rad(longitudes(i)-longitudes(1));

    a(i) = sin(difLat(i)/2)*sin(difLat(i)/2) + ...
```

```
        cos(deg2rad(latitudes(1)))*cos(deg2rad(latitudes(i)))* ...
        sin(difLon(i)/2)*sin(difLon(i)/2);
c(i) = 2*atan2(sqrt(a(i)),sqrt(1-a(i)));

dist(i) = R*c(i);

% erros:
erroXY_2(i) = abs(distXY_2(i)-dist(i))/dist(i)*100

end

[dist; distXY_2]
[dist; erroXY_2]
```

Referências

- [1] Isaac Skog e Peter Händel. In-car positioning and navigation technologies—a survey. *Intelligent Transportation Systems, IEEE Transactions on*, 10(1):4–21, 2009.
- [2] Elliott Kaplan e Christopher Hegarty. *Understanding GPS: principles and applications*. Artech house, 2005.
- [3] FX Breu, S Guggenbichler, e JC Wollmann. GLONASS Overview. *Vasa*, páginas 1–7, 2008. URL: <http://medcontent.metapress.com/index/A65RM03P4874243N.pdf>.
- [4] Tatiana Mirgorodskaya. Glonass government policy, status and modernization plans. Em *Proceedings of international global navigation satellite systems (IGNSS) symposium, Golden Coast*, páginas 16–18.
- [5] Chris A. Ciufu. Mems trends: Smaller, cheaper, everywhere, 2012. Acessado em: 2016-02-09. URL: <http://eecatalog.com/sensors/2012/05/24/mems-trends-smaller-cheaper-everywhere/>.
- [6] Paul D Groves. *Principles of GNSS, inertial, and multisensor integrated navigation systems*. Artech house, 2013.
- [7] Naser El-Sheimy, Haiying Hou, e Xiaoji Niu. Analysis and modeling of inertial sensors using allan variance. *IEEE Transactions on instrumentation and measurement*, 57(1):140–149, 2008.
- [8] NXP UM102104. I2c-bus specification and user manual, 2007.
- [9] VTI Technologies. Spi interface specification, 2005.
- [10] R. E. Nasburg e K. S. Moravec. Distributed multisensor data fusion. Em *AIAA/AHS/ASAE Aircraft Design Systems and Operations Meeting.*, AIAA Paper, páginas AIAA, New York, NY, USA; American Helicopter Soc, Washington, DC, USA; ASAE, Washington, DC, USA. AIAA. Compilation and indexing terms, Copyright 2015 Elsevier Inc. 1985020024394 BAYES DECISION THEORY DISTRIBUTED SENSOR FUSION EXECUTION ABILITY TOTAL-SYSTEM LIFE-CYCLE COSTS.
- [11] Julius Reiner. Application of expert systems to sensor fusion. Em *Proceedings of the IEEE 1985 National Aerospace and Electronics Conference, NAECON 1985.*, IEEE Proceedings of the National Aerospace and Electronics Conference, páginas 1444–1450. IEEE. Compilation and indexing terms, Copyright 2015 Elsevier Inc. 1986030040625 DATA FUSION LOGIC TREES.
- [12] Niklas Magnusson e Tobias Odenman. Improving absolute position estimates of an automotive vehicle using gps in sensor fusion. 2012.

- [13] Matias Tailanian, Santiago Paternain, Rodrigo Rosa, e Rafael Canetti. Design and implementation of sensor data fusion for an autonomous quadrotor. Em *2014 IEEE International Instrumentation and Measurement Technology Conference: Instrumentation and Measurement for Sustainable Development, I2MTC 2014, May 12, 2014 - May 15, 2014*, Conference Record - IEEE Instrumentation and Measurement Technology Conference, páginas 1431–1436. Institute of Electrical and Electronics Engineers Inc. Compilation and indexing terms, Copyright 2015 Elsevier Inc. 20143318066883 Accelerometer measurement Autonomous flight Autonomous platforms Design and implementations Design and integrations Inertial measurement unit State variables Systematic bias. URL: <http://dx.doi.org/10.1109/I2MTC.2014.6860982>, doi:10.1109/I2MTC.2014.6860982.
- [14] Andrew Mark Eldredge. Improved state estimation for miniature air vehicles. 2006.
- [15] Hang Guo, Min Yu, Chengwu Zou, e Wenwen Huang. Kalman filtering for gps/magnetometer integrated navigation system. *Advances in Space Research*, 45(11):1350–1357, 2010. URL: <http://www.sciencedirect.com/science/article/pii/S0273117710000463>, doi:<http://dx.doi.org/10.1016/j.asr.2010.01.011>.
- [16] Jeroen D Hol. Sensor fusion and calibration of inertial sensors, vision, ultra-wideband and gps. 2011.
- [17] Shaghayegh Zihajehzadeh, Darrell Loh, Tien Jung Lee, Reynald Hoskinson, e Edward J. Park. A cascaded kalman filter-based gps/mems-imu integration for sports applications. *Measurement*, 73:200–210, 2015. URL: <http://www.sciencedirect.com/science/article/pii/S0263224115002821>, doi:<http://dx.doi.org/10.1016/j.measurement.2015.05.023>.
- [18] S. Coraluppi e C. Carthel. Recursive track fusion for multi-sensor surveillance. *Information Fusion*, 5(1):23–33, 2004. URL: <http://www.sciencedirect.com/science/article/pii/S1566253503000666>, doi:<http://dx.doi.org/10.1016/j.inffus.2003.03.003>.
- [19] Shashank Shekhar, Oussama Khatib, e Makoto Shimojo. Sensor fusion and object localization. Em *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*, volume 3, páginas 1623–1628. IEEE.
- [20] Andrew Cavanaugh, Matthew Lowe, David Cyganski, e RJ Duckworth. Wpi precision personnel location system: Rapid deployment antenna system and sensor fusion for 3d precision location. Em *Proc., Institute of Navigation-International Technical Meeting 2010, Jan 25*, volume 27, páginas 384–389.
- [21] Ulrich Dieckmann, P Plankensteiner, Ralf Schamburger, Bernhard Fröba, e Sebastian Meller. Sesam: A biometric person identification system using sensor fusion. Em *Audio-and Video-based Biometric Person Authentication*, páginas 301–310. Springer.
- [22] Yasunari Yoshitomi, Sung Ill Kim, Takako Kawano, e T Kilazoe. Effect of sensor fusion for recognition of emotional states using voice, face image and thermal image of face. Em *Robot and Human Interactive Communication, 2000. RO-MAN 2000. Proceedings. 9th IEEE International Workshop on*, páginas 178–183. IEEE.
- [23] Steffen Knoop, Stefan Vacek, e Rüdiger Dillmann. Sensor fusion for 3d human body tracking with an articulated 3d body model. Em *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, páginas 1686–1691. IEEE.

- [24] LiveViewGPS. Gps and its three main competitors: Galileo, beidou, glonass, 2014. Acedido em: 2016-02-09. URL: <http://www.liveviewgps.com/blog/gps-main-competitors-galileo-beidou-glonass/>.
- [25] UNITED STATES NAVAL OBSERVATORY. Gps constellation status. Acedido em: 2016-02-09. URL: <ftp://tycho.usno.navy.mil/pub/gps/gpstd.txt>.
- [26] Mohinder S Grewal, Lawrence R Weill, e Angus P Andrews. *Global positioning systems, inertial navigation, and integration*. John Wiley & Sons, 2007.
- [27] Richard B Langley. Gps receiver system noise. *GPS world*, 8(6):40–45, 1997.
- [28] Russia Information-analytical centre, Korolyov. Glonass constellation status. Acedido em: 2016-02-09. URL: <https://www.glonass-iac.ru/en/GLONASS/>.
- [29] Keith M Miller. A review of glonass. *HYDROGRAPHIC JOURNAL*, páginas 15–22, 2000.
- [30] ESA. Space signal demonstrates galileo interoperability with gps, 2012. Acedido em: 2016-02-09. URL: http://www.esa.int/Our_Activities/Navigation/Space_signal_demonstrates_Galileo_interoperability_with_GPS.
- [31] Li Lingli. Beidou system commissioning services press conference held today, 2011. Acedido em: 2016-02-09. URL: <http://www.beidou.gov.cn/2011/12/27/201112273688a9fcf2494c73b3ca85aee280f86d.html>.
- [32] Daniel Goehl e David Sachs. Motion sensors gaining inertia with popular consumer electronics. *White Paper, IvenSense Inc*, 2007.
- [33] SP Reilly, RK Leach, A Cuenat, SA Awan, e M Lowe. Overview of mems sensors and the metrology requirements for their manufacture. *NPL Open Report DEPC-EM*, 8, 2006.
- [34] James Lenz e Alan S Edelstein. Magnetic sensors and their applications. *Sensors Journal, IEEE*, 6(3):631–649, 2006.
- [35] P Szendro, Gy Vincze, e A Szasz. Pink-noise behaviour of biosystems. *European Biophysics Journal*, 30(3):227–231, 2001.
- [36] Yongyao Cai, Yang Zhao, Xianfeng Ding, e James Fennelly. Magnetometer basics for mobile phone applications. *Electronic Products (Garden City, New York)*, 54(2), 2012.
- [37] Federico Castanedo. A review of data fusion techniques. *The Scientific World Journal*, 2013, 2013.
- [38] Venugopal V Veeravalli. Sequential decision fusion: theory and applications. *Journal of the Franklin Institute*, 336(2):301–322, 1999.
- [39] Bahador Khaleghi, Alaa Khamis, Fakhreddine O Karray, e Saiedeh N Razavi. Multisensor data fusion: A review of the state-of-the-art. *Information Fusion*, 14(1):28–44, 2013.
- [40] Tohru Katayama. *Subspace methods for system identification*. Springer Science & Business Media, 2006.
- [41] Mathieu St-Pierre e Denis Gingras. Comparison between the unscented kalman filter and the extended kalman filter for the position estimation module of an integrated navigation information system. Em *Intelligent Vehicles Symposium, 2004 IEEE*, páginas 831–835. IEEE.

- [42] Stefano Panzieri, Federica Pascucci, e Giovanni Ulivi. An outdoor navigation system using gps and inertial platform. *Mechatronics, IEEE/ASME Transactions on*, 7(2):134–142, 2002.
- [43] Adrian Smith, Arnaud Doucet, Nando de Freitas, e Neil Gordon. *Sequential Monte Carlo methods in practice*. Springer Science & Business Media, 2013.
- [44] Shoichi Maeyama, Akihisa Ohya, e Shin'ichi Yuta. Non-stop outdoor navigation of a mobile robot-retroactive positioning data fusion with a time consuming sensor system. Em *Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, volume 1, páginas 130–135. IEEE.
- [45] Yifeng Zhou, Henry Leung, e Patrick C Yip. An exact maximum likelihood registration algorithm for data fusion. *Signal Processing, IEEE Transactions on*, 45(6):1560–1573, 1997.
- [46] Pau Closas, Carles Fernández-Prades, e Juan A Fernández-Rubio. Maximum likelihood estimation of position in gnss. *Signal Processing Letters, IEEE*, 14(5):359–362, 2007.
- [47] Reza N Jazar. *Vehicle dynamics: theory and application*. Springer Science & Business Media, 2013.
- [48] Uwe Kiencke e Lars Nielsen. *Automotive control systems: for engine, driveline, and vehicle*, 2000.
- [49] Masato Abe. *Vehicle handling dynamics: theory and application*. Butterworth-Heinemann, 2015.
- [50] Chaiwat Nuthong. *Estimation of Tire-Road Friction Forces using Kalman Filtering for Advanced Vehicle Control*. Tese de doutoramento, München, Univ. der Bundeswehr, Diss., 2009, 2009.
- [51] Carlos Canudas-de Wit, Panagiotis Tsiotras, Efstathios Velenis, Michel Basset, e Gerard Gissinger. Dynamic friction models for road/tire longitudinal interaction. *Vehicle System Dynamics*, 39(3):189–226, 2003.
- [52] Eric A Wan e Rudolph Van Der Merwe. The unscented kalman filter for nonlinear estimation. Em *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, páginas 153–158. Ieee, 2000.
- [53] Simon J Julier e Jeffrey K Uhlmann. New extension of the kalman filter to nonlinear systems. Em *AeroSense'97*, páginas 182–193. International Society for Optics and Photonics, 1997.
- [54] Naren Naik, RMO Gemson, MR Ananthasayanam, et al. Introduction to the kalman filter and tuning its statistics for near optimal estimates and cramer rao bound. *arXiv preprint arXiv:1503.04313*, 2015.
- [55] T Fiorenzani, C Manes, G Oriolo, e P Peliti. Comparative study of unscented kalman filter and extended kalman filter for position/attitude estimation in unmanned aerial vehicles. *Inst. for Systems Analysis and Computer Science (IASI-CNR), Rome, Italy, Rept*, páginas 08–08, 2008.
- [56] Michael Roth e Fredrik Gustafsson. An efficient implementation of the second order extended kalman filter. Em *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*, páginas 1–6. IEEE, 2011.

- [57] Mikko Lehtinen, Ari Happonen, e Jouni Ikonen. Accuracy and time to first fix using consumer-grade gps receivers. Em *Software, Telecommunications and Computer Networks, 2008. SoftCOM 2008. 16th International Conference on*, páginas 334–340. IEEE, 2008.
- [58] Freescale Semiconductor. Allan variance: Noise analysis for gyroscopes, 2015.
- [59] CC Robusto. The cosine-haversine formula. *The American Mathematical Monthly*, 64(1):38–40, 1957.
- [60] John P Snyder. *Flattening the earth: two thousand years of map projections*. University of Chicago Press, 1997.
- [61] Michael F Goodchild e Sucharita Gopal. *The accuracy of spatial databases*. CRC Press, 1989.
- [62] Frédéric Leens. An introduction to i 2 c and spi protocols. *IEEE Instrumentation & Measurement Magazine*, 12(1):8–13, 2009.