



DISSERTAÇÃO

Mestrado em Engenharia Electrotécnica

# Condução Autónoma – Localização de um Veículo com base em Odometria, IMU e GPS

BAPTISTE FERREIRA GIL

Leiria, Março de 2014





DISSERTAÇÃO

Mestrado em Engenharia Electrotécnica

Energia e Automação

## **Condução Autónoma – Localização de um Veículo com base em Odometria, IMU e GPS**

BAPTISTE FERREIRA GIL

Dissertação de Mestrado realizada sob a orientação do Doutor Carlos Fernando Couceiro de Sousa Neves, Professor da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria e co-orientação do Doutor Hugo Filipe Costelha de Castro, Professor da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria.

Leiria, Março de 2014



*"We choose to go to the moon in  
this decade and do the other  
things, not because they are easy,  
but because they are hard..."*

**(John F. Kennedy),**  
*12 Settembre 1962*



# Agradecimentos

Este espaço é destinado aos agradecimentos das pessoas que contribuíram para o desenvolvimento desta dissertação de Mestrado. Deixo assim o meu reconhecimento e um muito obrigado para todos aqueles que, direta ou indiretamente, foram importantes para a conclusão desta etapa da minha vida.

Em primeiro lugar queria agradecer aos Professores Doutor Carlos Neves e Doutor Hugo Costelha, orientador e co-orientador desta dissertação respetivamente, a oportunidade e todo o apoio que me foi dado ao longo deste percurso.

Aos técnicos do laboratório de Engenharia Automóvel, Engenheiro Diogo Costa e Engenheiro Nuno Pires que sempre se mostraram interessados em ajudar no possível pela disponibilização do espaço de trabalho.

Aos seguranças do edifício D, principalmente ao senhor Gil, pela sua simpatia e compreensão.

Aos colegas de laboratório, Ricardo Santos, Miguel Rasteiro e todos aqueles que passavam por lá para ter novidades, pela vossa parceria e apoio prestado.

Aos parceiros de Intercâmbio, Cristóvão Rodrigues, João Gonçalves e Paulo Fernandes, pela vossa amizade e companheirismo demonstrada durante o curso, especialmente durante estes meses.

Ao Filipe Vieira, Tiago Lopes e Rui Lucas, amigos de longa data, pela vossa grande amizade e por todos os bons momentos passados ao longo destes anos.

À minha Família, em especial aos meus pais e irmãos, que sempre me apoiaram e acreditaram em mim.

A todos os mencionados, dedico este trabalho!



# Abstract

Driving a vehicle has become increasingly automated over the years. This process has been developed in order to facilitate the driving of vehicles, to prevent road accidents or to be used by the military. The main goal of this work is to study the two main methods used for probabilistic localization and their application in localizing an autonomous vehicle: the Extended Kalman Filter and the Particle Filter.

Several versions of both filters were implemented, in simulation and in a real vehicle, based on the robot odometry, an inertial unit and a GPS.

This work was the continuation of several previous projects on the development of a low cost autonomous vehicle, focusing on developing the localization system. In this sense, the work will serve as a basis for autonomous navigation in future work.

**Keywords:** Intelligent Vehicle, Vehicle Localization, Autonomous Driving, Extended Kalman Filter, Particle Filter, Sensor fusion.



# Resumo

A Condução de um veículo automóvel tem-se tornado cada vez mais automatizada ao longo dos anos. Este processo tem-se vindo a desenvolver com o intuito de facilitar cada vez mais a condução dos veículos, de prevenir acidentes rodoviários ou ainda de ser utilizado em operações militares. Este trabalho tem como principal foco o estudo da aplicação de dois dos principais métodos de cálculo da localização probabilística à localização de um veículo automóvel autónomo: o Filtro Estendido de Kalman e o Filtro de Partículas. Foram implementadas várias versões de ambos os filtros, quer em simulação, quer num veículo real, tendo por base, em termos de sensores, a odometria do robô, uma unidade inercial e um GPS.

Este trabalho surge na continuação de vários projetos anteriores, com foco no desenvolvimento de um veículo autónomo de baixo custo, acrescentando-lhe a funcionalidade de se localizar. Neste sentido, o trabalho desenvolvido servirá de base para a navegação autónoma em trabalhos futuros.

**Palavras-chave:** Veículos Inteligentes, Localização de Veículos, Condução Autónoma, Filtro Estendido de Kalman, Filtro de Partículas, Fusão sensorial.



# Índice

<b>Agradecimentos</b>	<b>iii</b>
<b>Abstract</b>	<b>v</b>
<b>Resumo</b>	<b>vii</b>
<b>Índice</b>	<b>xi</b>
<b>Lista de Figuras</b>	<b>xiii</b>
<b>Lista de Tabelas</b>	<b>xix</b>
<b>Lista de Abreviaturas</b>	<b>xxi</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Objetivos . . . . .	4
1.2 Organização . . . . .	4
<b>2 Metodologia de base</b>	<b>7</b>
2.1 Filtro de Kalman . . . . .	8
2.2 Filtro Estendido de Kalman . . . . .	9
2.2.1 Predição . . . . .	10
2.2.2 Correção . . . . .	11
2.3 Filtro de Partículas . . . . .	11
2.3.1 Predição . . . . .	13
2.3.2 Atualização . . . . .	13
2.3.3 Reamostragem . . . . .	14
<b>3 Modelos desenvolvidos</b>	<b>15</b>

3.1	Modelação do sistema . . . . .	15
3.1.1	Odometria do veículo . . . . .	16
3.1.2	Sensores inerciais . . . . .	18
3.1.3	GPS . . . . .	20
3.2	Método baseado no Filtro Estendido de Kalman . . . . .	24
3.2.1	EKF com odometria no passo de correção . . . . .	24
3.2.2	EKF com odometria no passo de predição . . . . .	27
3.3	Método baseado no Filtro de Partículas . . . . .	29
3.3.1	Predição . . . . .	29
3.3.2	Atualização . . . . .	30
3.3.3	Reamostragem . . . . .	31
<b>4</b>	<b>Implementação prática</b>	<b>33</b>
4.1	Sistema base . . . . .	33
4.2	Ambiente de Simulação . . . . .	34
4.3	Ambiente Real . . . . .	38
4.4	Implementação dos filtros . . . . .	41
4.4.1	EKF com a odometria no passo de correção . . . . .	41
4.4.2	EKF com a odometria no passo de predição . . . . .	43
4.4.3	Filtro de partículas . . . . .	43
<b>5</b>	<b>Resultados</b>	<b>47</b>
5.1	Testes realizados . . . . .	47
5.1.1	Ambiente de simulação . . . . .	49
5.1.2	Ambiente real . . . . .	70
<b>6</b>	<b>Conclusões e Trabalho Futuro</b>	<b>97</b>
	<b>Bibliografia</b>	<b>101</b>
<b>A</b>	<b>Caraterização dos sensores</b>	<b>105</b>
A.1	Sensores existentes . . . . .	105
A.1.1	Comunicação . . . . .	105
A.1.2	Caraterização do erro . . . . .	106
A.2	Sensores adicionados . . . . .	108

A.2.1	Comunicação . . . . .	109
A.2.2	Calibração e caracterização do erro . . . . .	110
A.2.3	Posicionamento dos sensores adicionados . . . . .	113
<b>B</b>	<b>Manual de utilizador</b>	<b>115</b>
B.1	SLVA . . . . .	115
B.1.1	Ambiente Simulação . . . . .	116
B.1.2	Ambiente Real . . . . .	116



# Lista de Figuras

1.1	<i>Alguns veículos autônomos desenvolvidos pela Google<sup>®</sup>.</i>	2
1.2	<i>Veículos autônomos desenvolvidos pela Vislab.</i>	3
1.3	<i>Atlascar - Veículo desenvolvido em Portugal.</i>	4
2.1	<i>Filtro de Kalman.</i>	9
2.2	<i>Trajectoria em que a incerteza decrescente é representada por uma nuvem de partículas mais concentrada em torno da trajetória real.</i>	12
3.1	<i>Sistema de eixos do veículo.</i>	17
3.2	<i>Deslocamento de um veículo entre dois instantes em escala exagerada.</i>	18
3.3	<i>Transformação de coordenadas geográficas para cartesianas [28].</i>	20
3.4	<i>Rotação de sistema de eixos.</i>	21
3.5	<i>Percurso medido do GPS, com e sem correção da orientação.</i>	23
3.6	<i>Diagrama para o processo de fusão sensorial.</i>	24
4.1	<i>Diagrama de comutação da aplicação entre ambiente de simulação e real.</i>	34
4.2	<i>Polaris Ranger EV - veículo utilizado para testes em simulação.</i>	35
4.3	<i>Tópicos de comunicação utilizados em ambiente de simulação.</i>	35
4.4	<i>Evolução de velocidade linear para diferentes ângulos do acelerador.</i>	36
4.5	<i>Controlador do tipo PI.</i>	36
4.6	<i>Volkswagen Sharan vr6.</i>	38
4.7	<i>Estrutura do sistema desenvolvido na dissertação anterior [14].</i>	39
4.8	<i>Ângulo obtido pelos sensores VS ângulo na convenção matemática</i>	40
4.9	<i>Tópicos de comunicação utilizados em ambiente real.</i>	40
4.10	<i>Intervalos de tempo entre aquisições sensoriais.</i>	41
4.11	<i>Exemplo da influência da distância no cálculo do peso das partículas.</i>	44

5.1	<i>Comparação do modelo PF com e sem estimação do estado na predição . . .</i>	48
5.2	<i>Erro do GPS, em posição, ao longo do percurso tipo elipse. . . . .</i>	50
5.3	<i>Erro do GPS, em orientação, ao longo do percurso tipo elipse. . . . .</i>	51
5.4	<i>Erro da odometria, em velocidade, ao longo do percurso tipo elipse. . . . .</i>	51
5.5	<i>Teste em simulação do tipo elipse, utilizando todos os sensores. . . . .</i>	52
5.6	<i>Evolução da posição em X ao longo do percurso tipo elipse. . . . .</i>	52
5.7	<i>Evolução da posição em Y ao longo do percurso tipo elipse. . . . .</i>	53
5.8	<i>Evolução da orientação ao longo do percurso tipo elipse. . . . .</i>	53
5.9	<i>Evolução da velocidade linear longo do percurso tipo elipse. . . . .</i>	53
5.10	<i>Erro de posição ao longo do percurso tipo elipse. . . . .</i>	54
5.11	<i>Erro de orientação ao longo do percurso tipo elipse. . . . .</i>	54
5.12	<i>Erro de velocidade ao longo do percurso tipo elipse. . . . .</i>	54
5.13	<i>Erro de posição ao longo do percurso tipo elipse tendo o GPS como referência. . . . .</i>	55
5.14	<i>Teste em simulação do tipo elipse, não utilizando a IMU. . . . .</i>	55
5.14	<i>Teste em simulação do tipo elipse, não utilizando a IMU. . . . .</i>	56
5.15	<i>Evolução da posição em X ao longo do percurso tipo elipse. . . . .</i>	56
5.16	<i>Evolução da posição em Y ao longo do percurso tipo elipse. . . . .</i>	56
5.17	<i>Evolução da orientação ao longo do percurso tipo elipse. . . . .</i>	57
5.18	<i>Evolução da velocidade linear ao longo do percurso tipo elipse. . . . .</i>	57
5.19	<i>Erro de posição ao longo do percurso tipo elipse. . . . .</i>	57
5.20	<i>Erro da orientação ao longo do percurso tipo elipse. . . . .</i>	58
5.21	<i>Erro da velocidade linear ao longo do percurso tipo elipse. . . . .</i>	58
5.22	<i>Erro de posição ao longo do percurso tipo elipse tendo o GPS como referência. . . . .</i>	59
5.23	<i>Erro do GPS em posição ao longo do percurso tipo oito. . . . .</i>	60
5.24	<i>Erro do GPS em orientação ao longo do percurso tipo oito. . . . .</i>	60
5.25	<i>Erro da odometria em velocidade linear ao longo do percurso tipo oito. . . . .</i>	61
5.26	<i>Teste em simulação do tipo oito. . . . .</i>	61
5.26	<i>Teste em simulação do tipo oito. . . . .</i>	62
5.27	<i>Evolução da posição em X ao longo do percurso tipo oito. . . . .</i>	62
5.28	<i>Evolução da posição em Y ao longo do percurso tipo oito. . . . .</i>	62
5.29	<i>Evolução da orientação ao longo do percurso tipo oito. . . . .</i>	63
5.30	<i>Evolução da velocidade linear ao longo do percurso tipo oito. . . . .</i>	63
5.31	<i>Erro de posição ao longo do percurso tipo oito. . . . .</i>	64

5.32	<i>Erro de orientação ao longo do percurso tipo oito.</i>	64
5.33	<i>Erro de velocidade ao longo do percurso tipo oito.</i>	64
5.34	<i>Erro de distância ao longo do percurso tipo oito tendo o GPS como referência.</i>	65
5.35	<i>Teste em simulação do tipo oito, não utilizando a IMU.</i>	65
5.36	<i>Evolução da posição em X ao longo do percurso tipo oito.</i>	66
5.37	<i>Evolução da posição em Y ao longo do percurso tipo oito.</i>	66
5.38	<i>Evolução da orientação ao longo do percurso tipo oito.</i>	66
5.39	<i>Evolução de velocidade linear ao longo do percurso tipo oito.</i>	67
5.40	<i>Erro de posição ao longo do percurso tipo oito.</i>	67
5.41	<i>Erro de orientação ao longo do percurso tipo oito.</i>	67
5.42	<i>Erro de velocidade ao longo do percurso tipo oito.</i>	68
5.43	<i>Erro de distância ao longo do percurso tipo oito tendo o GPS como referência.</i>	68
5.44	<i>Vista satélite da ESTG, via Google<sup>®</sup>.</i>	71
5.45	<i>Teste em ambiente real do tipo elipse, utilizando todos os sensores.</i>	71
5.46	<i>Evolução da posição em X ao longo do percurso tipo elipse.</i>	72
5.47	<i>Evolução da posição em Y ao longo do percurso tipo elipse.</i>	72
5.48	<i>Evolução da orientação ao longo do percurso tipo elipse.</i>	73
5.49	<i>Evolução da velocidade linear ao longo do percurso tipo elipse.</i>	73
5.50	<i>Erro de posição ao longo do percurso tipo elipse em ambiente real.</i>	73
5.51	<i>Erro de orientação ao longo do percurso tipo elipse em ambiente real.</i>	74
5.52	<i>Erro de velocidade ao longo do percurso tipo elipse em ambiente real.</i>	74
5.53	<i>Teste em ambiente real do tipo elipse, não utilizando a IMU.</i>	75
5.54	<i>Evolução da posição em X ao longo do percurso tipo elipse.</i>	75
5.55	<i>Evolução da posição em Y ao longo do percurso tipo elipse.</i>	76
5.56	<i>Evolução da orientação ao longo do percurso tipo elipse.</i>	76
5.57	<i>Evolução da velocidade ao longo do percurso tipo elipse.</i>	76
5.58	<i>Erro de posição ao longo do percurso tipo elipse.</i>	77
5.59	<i>Erro de orientação ao longo do percurso tipo elipse.</i>	77
5.60	<i>Erro de velocidade ao longo do percurso tipo elipse.</i>	77
5.61	<i>Teste em ambiente real do tipo oito.</i>	79
5.62	<i>Evolução da posição em X ao longo do percurso tipo oito.</i>	80
5.63	<i>Evolução da posição em Y ao longo do percurso tipo oito.</i>	80
5.64	<i>Evolução da orientação ao longo do percurso tipo oito.</i>	80

5.65	<i>Evolução da velocidade ao longo do percurso tipo oito.</i>	81
5.66	<i>Erro de distância ao longo do percurso tipo oito.</i>	81
5.67	<i>Erro da orientação ao longo do percurso tipo oito.</i>	81
5.68	<i>Erro da velocidade ao longo do percurso tipo oito.</i>	82
5.69	<i>Teste em ambiente real do tipo oito.</i>	82
5.70	<i>Teste em ambiente real do tipo oito.</i>	83
5.71	<i>Evolução da posição em X ao longo do percurso tipo oito.</i>	83
5.72	<i>Evolução da posição em Y ao longo do percurso tipo oito.</i>	84
5.73	<i>Evolução da orientação ao longo do percurso tipo oito.</i>	84
5.74	<i>Evolução da velocidade ao longo do percurso tipo oito.</i>	84
5.75	<i>Erro de distância ao longo do percurso tipo oito.</i>	85
5.76	<i>Erro da orientação ao longo do percurso tipo oito.</i>	85
5.77	<i>Erro da velocidade ao longo do percurso tipo oito.</i>	85
5.78	<i>Teste em ambiente real com utilização normal.</i>	87
5.79	<i>Evolução da posição em X ao longo do percurso de utilização normal.</i>	88
5.80	<i>Evolução da posição em Y ao longo do percurso de utilização normal.</i>	88
5.81	<i>Evolução da orientação ao longo do percurso de utilização normal.</i>	88
5.82	<i>Evolução da velocidade ao longo do percurso de utilização normal.</i>	89
5.83	<i>Erro de distância ao longo do percurso de utilização normal.</i>	89
5.84	<i>Erro da orientação ao longo do percurso de utilização normal.</i>	90
5.85	<i>Erro da velocidade ao longo do percurso de utilização normal.</i>	90
5.86	<i>Teste em ambiente real com utilização normal.</i>	91
5.87	<i>Evolução da posição em X ao longo de utilização normal.</i>	91
5.88	<i>Evolução da posição em Y ao longo de utilização normal.</i>	92
5.89	<i>Evolução da orientação ao longo de utilização normal.</i>	92
5.90	<i>Evolução da velocidade ao longo de utilização normal.</i>	92
5.91	<i>Erro de distância ao longo do percurso de utilização normal.</i>	93
5.92	<i>Erro da orientação ao longo do percurso de utilização normal.</i>	93
5.93	<i>Erro da velocidade ao longo do percurso de utilização normal.</i>	93
5.94	<i>Método EKF_2 no teste de utilização normal, com e sem IMU.</i>	95
A.1	<i>Ligação do MCBN ao PC.</i>	106
A.2	<i>Cadeia de caracteres correspondente à velocidade linear e ângulo direção.</i>	106

A.3	<i>Teste codificadores das rodas.</i>	107
A.4	<i>IMU - ref: MinIMU-9 v2.</i>	108
A.5	<i>GPS - ref: Holux GPSlim236.</i>	109
A.6	<i>Arduino Mega 2560 e respectivas ligações.</i>	110
A.7	<i>Calibração dos giroscópios.</i>	111
A.8	<i>Calibração de magnetómetros.</i>	112
A.9	<i>Velocidade linear da odometria versus GPS.</i>	112
A.10	<i>Pontos geodésicos utilizados para cálculo da variância do erro do GPS.</i>	113
A.11	<i>Posicionamento dos sensores adicionados.</i>	114
B.1	<i>Configurações do modelo a utilizar no SLVA.</i>	115
B.2	<i>Interface das aplicações.</i>	116



# Lista de Tabelas

2.1	<i>Definição de simbologia utilizada.</i>	8
2.2	<i>Processo de reamostragem do filtro de partículas. Sendo que os comentários estão inseridos dentro de chavetas "{comentário}".</i>	14
5.1	<i>Lista de testes efetuados.</i>	49
5.2	<i>O desvio padrão e média do erro dos sensores como referência, no teste tipo elipse.</i>	51
5.3	<i>Comparação entre algoritmos na simulação para o percurso do tipo elipse.</i>	59
5.4	<i>O desvio padrão e média do erro dos sensores como referência, no teste tipo oito.</i>	61
5.5	<i>Comparação entre algoritmos na simulação para o percurso do tipo oito.</i>	69
5.6	<i>Comparação entre algoritmos para todos os testes efetuados em simulação.</i>	69
5.7	<i>Comparação entre algoritmos em ambiente real para o percurso do tipo elipse.</i>	78
5.8	<i>Comparação entre algoritmos na simulação para o percurso do tipo oito.</i>	86
5.9	<i>Comparação entre algoritmos na simulação para o percurso de utilização normal.</i>	94
5.10	<i>Comparação entre algoritmos para todos os testes efetuados em ambiente real.</i>	95
A.1	<i>Especificações de comunicação dos sensores do veículo.</i>	105
A.2	<i>Variância e média do erro da velocidade linear.</i>	107
A.3	<i>Variância e média do erro do ângulo das rodas.</i>	108
A.4	<i>Especificações dos sensores inerciais - ref: MinIMU-9 v2.</i>	108
A.5	<i>Especificações do GPS - ref: Holux GPSlim236.</i>	109
A.6	<i>Especificações do Arduino Mega 2560.</i>	110
A.7	<i>Variância e média do erro dos acelerômetros.</i>	110
A.8	<i>Variância e média do erro da velocidade linear do GPS.</i>	113

A.9	<i>Variância e média do erro em posição, do GPS.</i>	113
B.1	<i>Processo para início da aplicação no Gazebo.</i>	117
B.2	<i>Processo para início da aplicação de atuação.</i>	117
B.3	<i>Processo para início de conexão <b>rosserial</b>.</i>	118
B.4	<i>Processo para início de conexão ao MCBN, sem compilar a aplicação.</i>	119

# Lista de Abreviaturas

**ABS** Sistema anti-bloqueio dos travões (*Anti-lock Braking System*)

**PF** Filtro de Partículas (*Particle Filter*)

**EKF** Filtro Estendido de Kalman (*Extended Kalman Filter*)

**pdf** função densidade de probabilidade (*probability density function*)

**2D** Bidimensional

**GPS** Sistema de Posicionamento Global (*Global Positioning System*)

**IMU** Unidade Inercial de Medida (*Inercial Measurement Unit*)

**cir** Centro instantâneo de rotação

**HDOP** Diluição de Precisão Horizontal (*Horizontal Dilution Of Precision*)

**ESTG** Escola Superior de Tecnologia e Gestão

**UKF** Filtro de Kalman *Unscented* (*Unscented Kalman Filter*)

**ROS** Sistema operativo do robô (*Robot Operating System*)

**MCBN** Módulo de Controlo de Baixo Nível.

**CEP** Erro Circular de Probabilidade (*Circular Error Probable*)

**VS** Contra (*Versus*)

**PI** Proporcional-Integral

**SID** Formato padrão de identificação de tramas *Standard IDentifier*

**SLVA** Sistema de Localização dum Veículo Automóvel



# Capítulo 1

## Introdução

Nos países industrializados, o veículo automóvel é uma realidade numa percentagem significativa da população que, segundo o *The World Bank*<sup>1</sup>, continua a aumentar. Consequentemente, também o número de acidentes rodoviários aumentou, sendo que a grande maioria destes deve a falhas humanas. Segundo [1], por ano, morrem mundialmente 1,2 milhões de pessoas e 50 milhões ficam feridas em acidentes rodoviários. Por forma a reduzir o número de acidentes são aplicadas cada vez mais, aos veículos automóveis, tecnologias de segurança (como por exemplo o ABS, o sistema de controlo de tração e os *airbags*) e sistemas de auxílio da condução (como por exemplo o sistema de travagem automática e o *cruise control* adaptativo). Espera-se que com a automatização da condução o número de acidentes diminua significativamente [2, 3].

Cada vez mais as pessoas recorrem ao auxílio das novas tecnologias nos veículos automóveis, quer por necessidade, quer apenas pelo aumento da qualidade e conforto na condução. O que no início foi desenvolvido sobretudo na área da robótica, e muitas vezes com fins militares, é cada vez mais utilizado em veículos comuns na vida quotidiana das pessoas. A condução autónoma introduz um conjunto de vantagens ao mercado [4], como o simples auxílio no estacionamento, a possibilidade de deslocação de pessoas com limitações físicas e/ou sensoriais e ainda a possibilidade de redução de distância entre veículos, conservando a sua segurança, tudo isto evitando falhas humanas. Tal implicaria um melhor aproveitamento das vias rodoviárias e a diminuição do trânsito, diminuindo assim também o consumo e a poluição efetuada pelos veículos.

Atualmente, existem no mercado inúmeros veículos com a capacidade de auxiliar a condução/manobras e até mesmo ajudar em caso de perigo. No caso dos veículos totalmente autónomos, não existem ainda modelos disponíveis no mercado. Existem contudo algumas soluções em fases de testes, com resultados importantes ao nível da condução autónoma. Talvez o mais conhecido e mais desenvolvido seja o veículo autónomo imple-

---

<sup>1</sup><http://data.worldbank.org/>

mentado pela Google<sup>®</sup> (ver Figura 1.1(a)). Este foi o primeiro carro autónomo a receber uma licença para circular em vias públicas, com a obrigação de ser supervisionado por duas pessoas, sendo uma delas o motorista para casos de emergência e a segunda para vigiar o computador que se encontra dentro da viatura [2]. Segundo o autor de [2], este veículo tem a capacidade de se conduzir autonomamente em 99% dos casos, em que o restante 1% são os casos imprevisíveis e mais perigosos, que causariam acidentes mesmo sendo humanos a conduzir.

O algoritmo de localização utiliza informação sensorial com base na odometria, na câmara de vídeo, no Lidar (sensor laser rotacional), no GPS e nos diversos radares em torno do veículo. De salientar que o Lidar utilizado tem um valor comercial na ordem dos \$75000<sup>2</sup>, o que torna esta abordagem inviável em termos comerciais, pelo menos considerando uma distribuição em larga escala.

Não existe muita informação disponível, mas supõe-se que o modelo de fusão sensorial utilizado seja semelhante ao modelo utilizado no veículo *Stanley*, adaptado para ambiente urbano. Descrito em [5], o modelo utilizado no *Stanley* foi baseado no filtro de Kalman *Unscented* (UKF) que, segundo os autores, em média, tem um erro de estimação inferior, comparando por exemplo com o filtro estendido de Kalman (EKF).

*Stanley* (ver Figura 1.1(b)) é um veículo desenvolvido, pela *Stanford Racing Team* da Universidade de Stanford (Estados Unidos), para condução autónoma em ambiente *outdoor*, tendo como principal objetivo a competição nas provas *Grand Challenge*, organizadas pela *Defense Advanced Research Projects Agency (DARPA)*, descritas em [6], tendo sido vencedor no *DARPA Grand Challenge 2005*. O algoritmo de localização (UKF) utiliza informação sensorial com base na odometria, na câmara de vídeo, no telémetro a laser, no GPS, nos sensores inerciais e nos diversos radares em torno do veículo.



(a) Veículo desenvolvido para ambiente urbano.



(b) Veículo desenvolvido para ambiente outdoor - Stanley.

Figura 1.1: Alguns veículos autónomos desenvolvidos pela Google<sup>®</sup>.

<sup>2</sup><http://www.popsci.com/cars/article/2013-09/google-self-driving-car>

Desenvolvidos pela VisLab (*Vision Laboratory*) para a prova *VisLab Intercontinental Autonomous Challenge* (VIAC), estes veículos (quatro unidades semelhantes à Figura 1.2(a)) percorreram 13.000 km em 3 meses, desde Parma (Itália) até Xangai (China), no modo de condução autónoma. Para além de serem veículos de condução completamente autónoma, são veículos elétricos. Na figura destaca-se o painel solar, que visa a alimentação da parte de controlo do veículo, uma opção da equipa de construção, com o intuito de mitigar a influência deste módulo na autonomia do veículo. A informação sensorial utilizada nestes veículos, para o algoritmo de localização, é baseada na odometria, na câmara de vídeo, no telémetro a laser, no GPS, e nos sensores inerciais.

Mesmo existindo muita informação disponível, não foi encontrada a informação relativa ao modelo de fusão sensorial utilizado, mas supõe-se que tenha sido usado um modelo semelhante aos veículos de base [7], nomeadamente o ARGO [8], o TerraMax [9] e o BRAiVE [10], ambos desenvolvidos pela Vislab. Segundo [11], o modelo de fusão sensorial utilizado para o TerraMax foi baseado no EKF.

BRAiVE [10] (ver Figura 1.2(b)) foi um dos últimos veículos desenvolvido pela Vislab, e tem o objetivo de competir nas provas *DARPA Urban Challenge*, descritas em [6]. Tem a particularidade de, à primeira vista, ser um veículo comum visto, de acordo com as regras desta prova, ter os sensores incorporados no próprio veículo. O algoritmo de localização utiliza informação sensorial com base nas mesmas utilizadas no veículo desenvolvido para as provas VIAC [10], adaptadas para o ambiente das provas em questão.



(a) Veículo utilizado no VIAC.



(b) BRAiVE.

Figura 1.2: *Veículos autónomos desenvolvidos pela Vislab.*

O Atlascar<sup>3</sup> é um outro projeto de condução autónoma de baixo custo, comparando com os outros veículos apresentados. Este foi criado e desenvolvido pelo Departamento de Engenharia Mecânica da Universidade de Aveiro - Portugal, com base em outros robôs de menor escala [12], utilizados para competir em provas de condução autónoma, os quais foram premiados por várias vezes no Festival Nacional de Robótica. A localização neste protótipo foi efetuada com base nos sensores inerciais (IMU), no GPS, num laser 2D e

<sup>3</sup><http://atlas.web.ua.pt/atlascar.html>

3D, na câmara estéreo, na câmara térmica e nos dois lasers na sua frente, tal como mostra a Figura 1.3. Neste caso, os autores optaram por utilizar o Filtro Estendido de Kalman [13] como base para o modelo de fusão sensorial.



Figura 1.3: *Atlascar - Veículo desenvolvido em Portugal.*

Para que um veículo automóvel possa ter um sistema de condução autónoma, necessita dum sistema para planeamento de trajetórias que, por sua vez, implica que possua um processo de localização. Tendo por base uma dissertação desenvolvida anteriormente [14], onde foi desenvolvido o controlo dum veículo automóvel através do acoplamento de atuadores, sensores e da eletrónica necessária para o seu controlo, esta dissertação foca-se no desenvolvimento do processo de localização deste veículo. Na secção seguinte, são explicados em detalhe os objetivos pretendidos para esta dissertação.

## 1.1 Objetivos

Esta dissertação tem como intuito fazer o estudo da aplicação de dois dos principais métodos de cálculo da localização probabilística à localização de um veículo automóvel autónomo: o Filtro Estendido de Kalman e o Filtro de Partículas. Foram implementadas várias versões de ambos os filtros, quer em simulação, quer num veículo real, tendo por base, em termos de sensores, a odometria do robô, uma unidade inercial e um GPS.

Pretende-se com este projeto desenvolver um sistema de localização de baixo custo para um veículo automóvel com o intuito de, posteriormente, se incluir num veículo para condução autónoma. A robustez do processo de localização é um fator fundamental para a realização daquele objetivo, sendo um dos principais fatores no planeamento de rotas.

## 1.2 Organização

Neste capítulo foi feita uma breve introdução sobre a condução autónoma, expondo-se também alguns casos de sucesso. Segue-se a definição de fusão sensorial e a justificação da sua utilização, no Capítulo 2, apresentando-se também alguns dos principais métodos

utilizados para o efeito, nomeadamente o filtro estendido de Kalman e o filtro de partículas (PF).

No Capítulo 3, é feita a adaptação dos métodos de fusão sensorial para os modelos em estudo, analisados no capítulo anterior e, após o estudo dos ambientes para realização de testes (ambiente de simulação e ambiente real), no Capítulo 4 é feita a adaptação destes métodos de fusão sensorial segundo as características destes ambientes. No Capítulo 5, são apresentados resultados dos testes efetuados nos vários ambientes, incluindo uma análise comparativa. Por fim, são apresentadas as conclusões finais, aproveitando para sugerir algumas considerações a implementar no futuro, no Capítulo 6.

Em Apêndice é apresentada a caracterização dos sensores, nomeadamente as principais características, a calibração implementada, bem como os esquemas de comunicação existentes (Apêndice A). É ainda disponibilizado o manual do utilizador, Apêndice B, apresentando os passos necessários para correr as aplicações utilizadas, tanto em ambiente de simulação como em ambiente real.



# Capítulo 2

## Metodologia de base

Para que um veículo automóvel possa ter um sistema de condução autónoma, é essencial que possua um sistema de localização. O sistema de localização é crucial e contribui significativamente para a autonomia do sistema.

Dum modo geral, a localização pode dividir-se em localização relativa e localização absoluta [15]. A localização relativa é obtida com base no estado anterior, integrando deslocamentos ou velocidades, por exemplo, enquanto que a localização absoluta é obtida em relação a uma referência considerada fixa, por exemplo, uma referência global associada ao centro da Terra.

Dependendo do tipo e qualidade do sensor e da função desejada, poderá ser utilizado apenas um ou múltiplos sensores para a localização. Do ponto de vista económico, pode ser mais aliciante a utilização de vários sensores de baixo custo, tipicamente de menor precisão, do que a utilização de apenas um sensor com grande precisão mas de custo mais elevado. A utilização de múltiplos sensores implica a fusão da informação dos mesmos, cujo intuito é de usufruir das vantagens de cada sensor e de atenuar as suas desvantagens.

A fusão sensorial [16] consiste na combinação da informação de diferentes sensores para obter uma estimativa melhor do que a que se obteria considerando cada sensor individualmente. Também permite obter informação que não é disponibilizada a partir de um único sensor. Qualquer sensor tem um erro associado que depende do tipo e qualidade do mesmo. A conjugação de informações de diferentes sensores permite minimizar o erro da estimativa global. A combinação de múltiplos sensores tem também a vantagem do sistema ser mais preciso e mais robusto, comparando com cada sensor individualmente. Assim, no caso de ocorrer uma falha num dos sensores, o erro associado poderá ser atenuado pelos outros sensores, resultando assim na minimização do erro à saída do sistema.

Os sensores devolvem medidas representantes da evolução do sistema, medidas essas que, na prática, são sempre afetadas por ruído, o qual se assume aleatório. Torna-se assim natural modelar estes sistemas como processos estocásticos. Existem diversas técnicas de

análise de processos estocásticos (função/processo temporal que varia aleatoriamente), como o filtro de partículas, o filtro de Kalman e filtros derivados do filtro de Kalman [17]. Estas técnicas serão discutidas mais em detalhe nas secções seguintes. Para uma melhor interpretação destas técnicas, será sempre utilizada a mesma simbologia, definida na Tabela 2.1.

Tabela 2.1: *Definição de simbologia utilizada.*

Símbolos	Descrição
$X$	Estado
$u$	Sinal de controlo
$z$	Vetor de medidas
$h()$	Função não linear das medidas
$f()$	Função não linear
$k$	Tempo/Instante anterior
$k + 1$	Tempo/Instante atual
$\hat{\phantom{x}}$	Estimativa
$-$	Após predição/Não corrigido

## 2.1 Filtro de Kalman

Desenvolvido em 1960 por Rudolf Kalman [18], o Filtro de Kalman (KF) [19, 20] é um eficiente filtro recursivo, que estima o estado de um sistema dinâmico linear a partir de uma série de medições ruidosas, minimizando o erro quadrático de estimação. Quando respeitadas as condições normais de funcionamento, isto é, quando o sistema é linear e o erro associado é Gaussiano de média nula, o filtro é ótimo.

Conciliando a informação de diferentes sensores (fusão sensorial) que forneçam informações relacionadas entre si, direta ou indiretamente, é possível estimar o estado do sistema. Para que tal resulte, e tendo em conta o modelo do ruído associado a cada sensor, é atribuída uma incerteza a cada medida.

Este filtro divide-se em dois passos: predição e correção. No passo de predição, é efetuada uma estimativa atual do estado tendo por base o estado anterior e os sinais de controlo/atuação. Neste passo, também é atualizada a incerteza da estimativa do estado. O passo de correção pode ser dividido em duas fases: observação e atualização. Na fase de observação, é determinado o erro entre as medidas obtidas pelos sensores e as calculadas em função do estado estimado na predição. Na fase de atualização, é feita a correção do estado, tendo em consideração o erro observado e a confiança associada às medidas. Os passos de predição e correção são repetidos para cada iteração, tendo como base o

resultado anterior (Figura 2.1), resultando assim num processo recursivo.

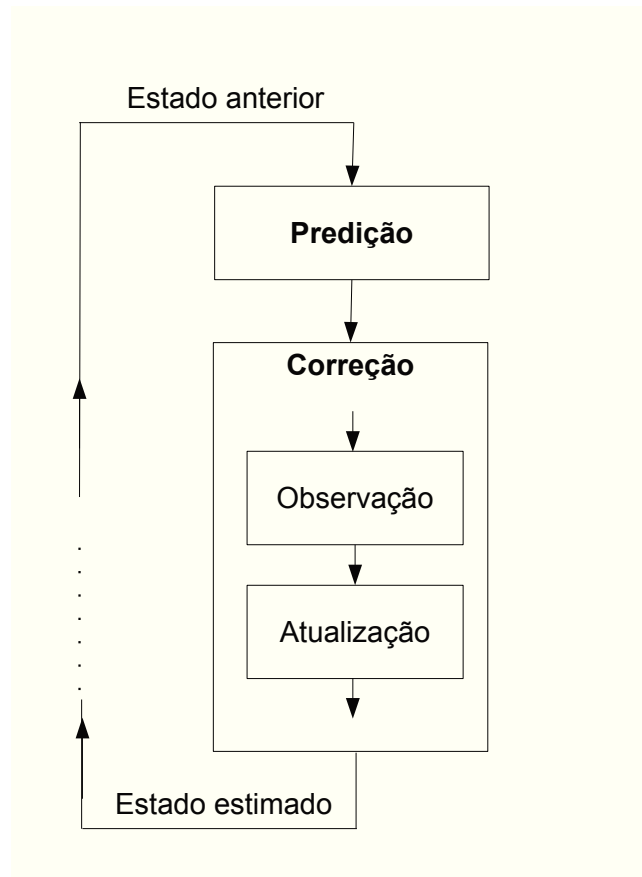


Figura 2.1: *Filtro de Kalman*.

## 2.2 Filtro Estendido de Kalman

O Filtro Estendido de Kalman (EKF) é uma derivação do Filtro de Kalman para aplicação em sistemas não lineares, considerando a linearização do sistema em torno da última estimativa. Não obstante a utilização generalizada com sucesso do EKF, esta aproximação leva a que o resultado não seja ótimo, podendo inclusive não convergir para a solução correta.

Na aplicação do EKF assume-se que o sistema é um processo estocástico, com o seguinte modelo de estado:

$$X_{k+1} = f(X_k, u_k, \gamma_k, \sigma_k) \quad (2.1)$$

Onde  $X_{k+1}$  representa o estado para o instante  $k + 1$ ;  $f$  é uma função que representa a dinâmica do estado não linear;  $X_k$  representa o vetor de estado anterior (instante  $k$ );  $u_k$  representa o vetor de sinal de controlo para o instante  $k$ ;  $\gamma_k$  e  $\sigma_k$  representam o ruído

da dinâmica do estado e da entrada, respetivamente.

Dado um modelo de sensores, o vetor de medidas esperado para um dado estado é representado pela seguinte equação:

$$z_{k+1} = h(X_{k+1}, v_{k+1}) \quad (2.2)$$

Aqui  $z_{k+1}$  representa o vetor de medidas;  $h$  é uma função possivelmente não linear; e  $v_{k+1}$  o ruído de medida.

Com base nas equações e pressupostos indicados acima, o EKF permite obter uma estimativa do estado com base nos passos indicados anteriormente (para o caso do KF), nomeadamente:

- **Predição** - Estimação do estado e covariância atuais tendo por base o modelo da dinâmica do estado.
- **Correção** - Correção do valor estimado no passo de predição utilizando a informação do vetor de medidas.

Estes passos estão descritos de uma forma mais detalhada na Secção 2.2.1 e na Secção 2.2.2, respetivamente.

### 2.2.1 Predição

O estado atual e a respetiva covariância são estimadas usando o modelo da dinâmica de estado definido pela Equação (2.1), considerando a melhor estimativa possível, isto é, considerando ruído nulo [21]. Tal resulta na Equação (2.3) e Equação (2.4), respetivamente:

$$\hat{X}_{k+1}^- = f(\hat{X}_k, u_k) \quad (2.3)$$

$$P_{k+1}^- = A_{k+1}P_kA_{k+1}^T + B_{k+1}\Gamma_kB_{k+1}^T + Q_k \quad (2.4)$$

Onde  $\hat{X}_{k+1}^-$  representa o estado estimado, não corrigido, para o instante  $k + 1$ ;  $P_{k+1}^-$  corresponde à matriz de covariância associada à previsão do estado, não corrigida, no instante  $k + 1$ ;  $P_k$  corresponde à matriz de covariância do estado corrigida para o instante  $k$ ;  $\Gamma_k$  corresponde à matriz de covariância das entradas;  $Q_k$  corresponde à matriz de covariância do ruído da dinâmica do sistema;  $A_{k+1}$  e  $B_{k+1}$  correspondem às matrizes do jacobiano da dinâmica do sistema em ordem a  $X$  e  $u$ , respetivamente – Equação (2.5) e Equação (2.6).

$$A_{k+1} = \left. \frac{\partial f}{\partial X} \right|_{\substack{X=\hat{X}_k \\ u=\hat{u}_k}} \quad (2.5)$$

$$B_{k+1} = \left. \frac{\partial f}{\partial u} \right|_{\substack{X=\hat{X}_k \\ u=\hat{u}_k}} \quad (2.6)$$

### 2.2.2 Correção

O estado atual é corrigido em função das medidas adquiridas pelos sensores. Após atualização da importância associada aos sensores (ganho de Kalman) é corrigido o estado estimado e a sua covariância [21] - Equação (2.7), Equação (2.8) e Equação (2.9) respectivamente.

$$K_{k+1} = P_{k+1}^- H_{k+1}^T (H_{k+1} P_{k+1}^- H_{k+1}^T + R_{k+1})^{-1} \quad (2.7)$$

$$\hat{X}_{k+1} = \hat{X}_{k+1}^- + K_{k+1} (z_{k+1} - h(\hat{X}_{k+1}^-)) \quad (2.8)$$

$$P_{k+1} = (I - K_{k+1} H_{k+1}) P_{k+1}^- \quad (2.9)$$

Aqui  $K_{k+1}$  corresponde ao ganho de Kalman no instante  $k+1$ ;  $\hat{X}_{k+1}$  ao estado estimado corrigido no instante  $k+1$ ;  $R_{k+1}$  corresponde à matriz de covariância do ruído dos sensores no instante  $k+1$ ;  $I$  é a matriz identidade; e  $H_{k+1}$  corresponde à matriz do jacobiano das medidas relativamente ao estado no instante  $k+1$ :

$$H_{k+1} = \left. \frac{\partial h}{\partial X} \right|_{X=\hat{X}_{k+1}^-} \quad (2.10)$$

## 2.3 Filtro de Partículas

O filtro de partículas (PF) é um outro método de estimativa do estado de um determinado sistema, baseado no filtro de Bayes. O filtro de Bayes consiste na atualização de uma função densidade de probabilidade (pdf) posterior do estado, baseando-se no estado atual e na leitura atual dos sensores, através da Regra de Bayes [22, 23, 24]. O filtro de partículas baseia-se na utilização de um conjunto de estimativas do estado, sendo cada uma destas estimativas designada por partícula [17]. Este é composto por três processos: predição, atualização e reamostragem. A predição consiste na alteração de valor de todas as partículas com base no modelo da dinâmica de estado, considerando ruído associado a cada partícula de forma independente. Cada partícula tem um peso associado que corresponde à sua confiança. Este peso é definido no processo de atualização, tendo por base a diferença entre o valor obtido a partir dos sensores e o valor esperado das medidas, considerando o estado associado à partícula correspondente. Quanto menor for essa diferença, maior será a confiança associada a essa partícula, ou seja, maior será o

peso dessa partícula. Por fim, a reamostragem consiste na replicação das partículas, com uma probabilidade igual ao seu peso. A longo prazo, esta operação resulta efetivamente na replicação das partículas com maior peso e no descarte daquelas com menor peso, resultando assim numa densidade da nuvem de partículas mais concentrada em torno das partículas de maior peso.

Na Figura 2.2 está representada uma trajetória efetuada com diversas iterações do filtro, onde é possível constatar que a função densidade de probabilidade, representada por um conjunto de  $N$  amostras (partículas) a azul, vai se tornando cada vez mais concentrada em torno do valor real do estado.

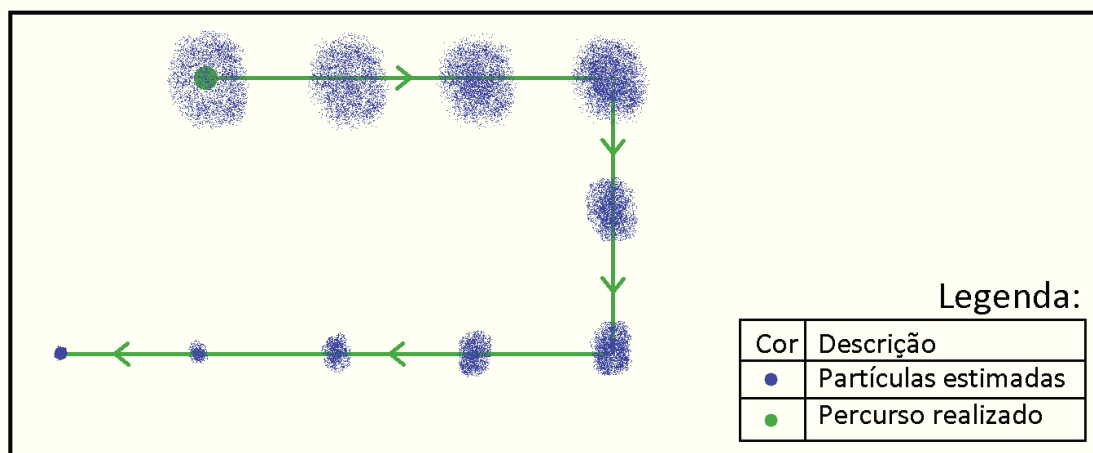


Figura 2.2: Trajetória em que a incerteza decrescente é representada por uma nuvem de partículas mais concentrada em torno da trajetória real.

À semelhança do EKF, assume-se que o processo a estimar é descrito por um estado ( $\hat{X}_{k+1}$ ), uma função dinâmica do estado ( $f$ ) e uma função que modela os sensores ( $h$ ) - Equação 2.11 e Equação 2.12, respetivamente.

$$X_{k+1} = f(X_k, u_k, w_k) \quad (2.11)$$

$$z_{k+1} = h(X_{k+1}, v_{k+1}) \quad (2.12)$$

Onde  $X_{k+1}$  continua a ser o estado estimado para o instante  $k+1$ ;  $f$  é uma função não linear estocástica;  $X_k$  o vetor de estado anterior corrigido, para o instante  $k$ ;  $u_k$  o vetor de sinais de controlo no instante  $k$ ;  $w_k$  corresponde a um ruído aleatório no instante  $k$ ;  $z_{k+1}$  o vetor de medidas (valores dos sensores) no instante  $k+1$ ;  $h$  uma função possivelmente não linear que modela o comportamento dos sensores; e  $v_{k+1}$  um ruído aleatório referente às medida no instante  $k+1$ . Estas equações aplicam-se a cada partícula, sendo o estado específico para cada uma e o vetor de medidas e o de sinais de controlo comuns entre elas. Neste caso, não é necessário que o modelo seja linear, nem é necessário que o ruído

seja Gaussiano de média nula [17, 22]. No entanto, este método também não fornece as garantias que o filtro de Kalman proporciona, ou seja, não garante ser ótimo.

### 2.3.1 Predição

No processo de predição são gerados estados hipotéticos (partículas) a partir do conjunto de partículas anterior, dos sinais de controlo e de ruído aleatório, com base na Equação 2.11. Cada partícula pode ser representada pela Equação (2.13).

$$S_{k+1}^{(i)} = (\hat{X}_{k+1}^{(i)-}, \pi_k^{(i)}) \quad (2.13)$$

Em que  $i$  corresponde ao índice da partícula, variando entre 1 e  $N$ ;  $S_{k+1}^{(i)}$  corresponde à partícula  $i$ , sendo composta por um estado e um peso no instante  $k + 1$ ;  $\hat{X}_{k+1}^{(i)-}$  representa a previsão do estado da partícula  $i$  no instante  $k + 1$ , definido pela Equação 2.11;  $\pi_k^{(i)}$  representa o peso da partícula  $i$  no instante  $k$ .

### 2.3.2 Atualização

À semelhança da fase de observação no filtro estendido de Kalman, no processo de atualização é feita uma comparação entre as medidas obtidas e as medidas esperadas. Estas últimas são estimadas através da Equação 2.12.

Em função da comparação efetuada, é atribuído um peso proporcional a cada partícula [17], ver Equação (2.14). Este peso corresponde à confiança atribuída a cada partícula, em que quanto maior este for, maior será a confiança e vice-versa, ou seja, segue o modelo de uma função densidade de probabilidade (pdf).

$$\pi_{k+1}^{(i)'} = \frac{1}{\sqrt{(2\pi)^N |W_{k+1}|}} e^{-\frac{1}{2}(z-\hat{z})^T W_{k+1}^{-1} (z-\hat{z})} \quad (2.14)$$

Onde  $\pi_{k+1}^{(i)'}$  representa o peso da partícula  $i$ , não normalizado, no instante  $k + 1$ ;  $W_{k+1}$  corresponde à matriz da covariância definida para a leitura dos sensores.

Tendo os pesos atualizados, é feita a sua normalização para garantir que estes representam uma função densidade de probabilidade com soma igual a 1. Esta normalização consiste na divisão do peso de cada partícula pelo peso total de todas as partículas [17], ver Equação (2.15).

$$\pi_{k+1}^{(i)} = \frac{\pi_{k+1}^{(i)'}}{\sum_{i=1}^N \pi_{k+1}^{(i)'}} \quad (2.15)$$

### 2.3.3 Reamostragem

Na reamostragem é feita a replicação das partículas com probabilidade proporcional ao peso da partícula, de modo a obter uma densidade da nuvem de partículas mais concentrada em torno das partículas com maior confiança. Existem vários métodos utilizados para este passo, sendo um deles representado em pseudocódigo na Tabela 2.2 [22].

```

 $n_{aleatorio} = aleatorio[0; 1/N]$    {Gera número aleatório entre 0 e 1/N}
 $c = \pi_k^{(1)}$ 
 $i = 1$ 
para  $j = 1$  até  $N$ , fazer   {Para cada partícula}
     $u = n_{aleatorio} + j/N$ 
    enquanto  $u > c$    {Enquanto a probabilidade não é proporcional ao peso}
         $i = i + 1$ 
         $c = c + \pi_k^{(i)}$ 
    fim do enquanto
     $X_{k+1}^{(j)} = X_k^{(i)}$    {Replica partícula}
     $\pi_{k+1}^{(j)} = \pi_k^{(i)}$    {Atribui o peso respectivo}
fim do fazer

```

Tabela 2.2: *Processo de reamostragem do filtro de partículas. Sendo que os comentários estão inseridos dentro de chavetas "{comentário}"*.

Após a replicação, o peso é reiniciado para todas as partículas, voltando a ter o valor inicial de  $(\pi_{k+1}^{(i)} = 1/N)$ , ou seja, nesta etapa o peso é distribuído de igual forma para todas as partículas. Pode-se, por isso, considerar que as partículas são definidas pela Equação (2.16).

$$S_{k+1}^{(i)} = \left( X_{k+1}^{(i)}, \frac{1}{N} \right) \quad (2.16)$$

# Capítulo 3

## Modelos desenvolvidos

Neste capítulo descrevem-se os modelos desenvolvidos para a localização de um veículo automóvel, baseados nos filtros descritos no Capítulo 2. Numa primeira fase, é feita a modelação do sistema, descrevendo o porquê da utilização de cada sensor e da fusão sensorial destes, seguindo-se a descrição dos modelos desenvolvidos. O primeiro modelo representado baseia-se no EKF. Este é apresentado em duas implementações distintas, sendo uma utilizando a odometria como um sensor comum no processo de atualização e outra utilizando a odometria na predição, considerando esta como um sinal de controlo. O segundo modelo representado é baseado no PF.

### 3.1 Modelação do sistema

Tal como descrito anteriormente, optou-se por uma abordagem probabilística na estimativa da pose do veículo, tendo por base o modelo da dinâmica do estado e múltiplos sensores. Para tal foi feita a fusão sensorial de três tipos de sensores: odometria, unidade inercial e GPS.

O GPS é frequentemente utilizado para funções de localização devido a ser um sensor de localização absoluta, podendo assim localizar-se no espaço, mesmo desconhecendo o estado anterior. A precisão destes sensores varia bastante consoante a qualidade e o tipo de sensor GPS, estando tipicamente entre 1 cm a vários metros. Em contrapartida, este sensor tem a desvantagem de ter uma baixa frequência de aquisição de dados (tipicamente entre 1 Hz e 3 Hz) e de perder a precisão, ou ainda, na pior das situações, deixar de funcionar com alguns fatores externos. Um exemplo de fator prejudicial para este tipo de sensor seria a passagem num túnel ou condições climatéricas desfavoráveis.

Para corrigir/reduzir as desvantagens associadas ao GPS, é utilizada também a odometria. Esta fonte de informação sensorial é composta, no caso típico de um veículo au-

tomóvel, por codificadores acoplados às rodas e por um codificador acoplado à coluna de direção (ângulo do volante que se traduz, indiretamente, no ângulo das rodas direcionais).

A odometria tem a vantagem de ter uma alta frequência de aquisição, o que reduz os instantes sem qualquer informação sensorial. As duas principais razões para não poder utilizar unicamente este sensor são: por este ser um meio de efetuar localização relativa, ou seja, é necessário conhecer o estado anterior para prever o estado atual, logo terá erro cumulativo; a sua precisão ser influenciada por fatores externos, como o raio das rodas (variação na pressão dos pneus) ou derrapagem efetuada pelas mesmas.

A pensar na utilização futura num espaço tridimensional (3D) foram acoplados também sensores inerciais (acelerómetros e giroscópios) e magnetómetros, a este conjunto, acelerómetros, giroscópios e magnetómetros, é denominado IMU (*Inertial Measuring Unit*). Neste trabalho assumiu-se uma situação inicial simplificada, em que o veículo se desloca aproximadamente num plano, pelo que a utilização de um modelo bidimensional (2D) é suficiente. Sendo assim, recorreu-se apenas a um eixo dos acelerómetros (horizontal, alinhado com o veículo) e dos giroscópios (eixo vertical) e ao magnetómetro, não obstante os três sensores disponibilizarem informação em três eixos. À semelhança da odometria, a IMU também tem alta frequência de aquisição de dados, apresentando a vantagem de permitir obter a orientação absoluta. O cálculo da posição (relativa) é realizado através da integração dupla da aceleração, tendo por isso erro cumulativo. As medidas fornecidas pela IMU são afetadas por fatores externos, nomeadamente trepidações e campos magnéticos. Dados estes fatores, a utilização apenas deste sensor também não seria uma boa opção.

Assumindo um veículo equipado com os três tipos de sensores referidos (GPS, IMU e odometria), para poder estimar o estado atual é necessário ter em conta o estado anterior e o deslocamento efetuado desde este. Sendo usado um modelo 2D, o deslocamento linear será considerado ao longo de dois eixos (X e Y) e o deslocamento angular ( $\theta$ ) em torno de (Z), tal como descrito na secção seguinte.

### 3.1.1 Odometria do veículo

Através de codificadores acoplados às rodas e ao volante é calculada a odometria. Destes sensores é extraída a velocidade linear do veículo (no sentido direcional de  $X'$  do sistema de eixos do veículo, ver Figura 3.1) e o ângulo das rodas direcionais. O veículo utiliza a direção de *Ackerman* podendo ser modelado como um veículo de duas rodas [25].

Na Figura 3.1 está representado, a preto, um veículo típico com direção de *Ackerman*, com o sistema de eixos local ( $X'$ ,  $Y'$  e  $Z'$ ) e o global (X, Y e Z) representados a vermelho, o centro instantâneo de rotação (cir) representado por um ponto verde e as rodas virtuais em azul. Também a verde, está representada a orientação da velocidade linear do veículo e o

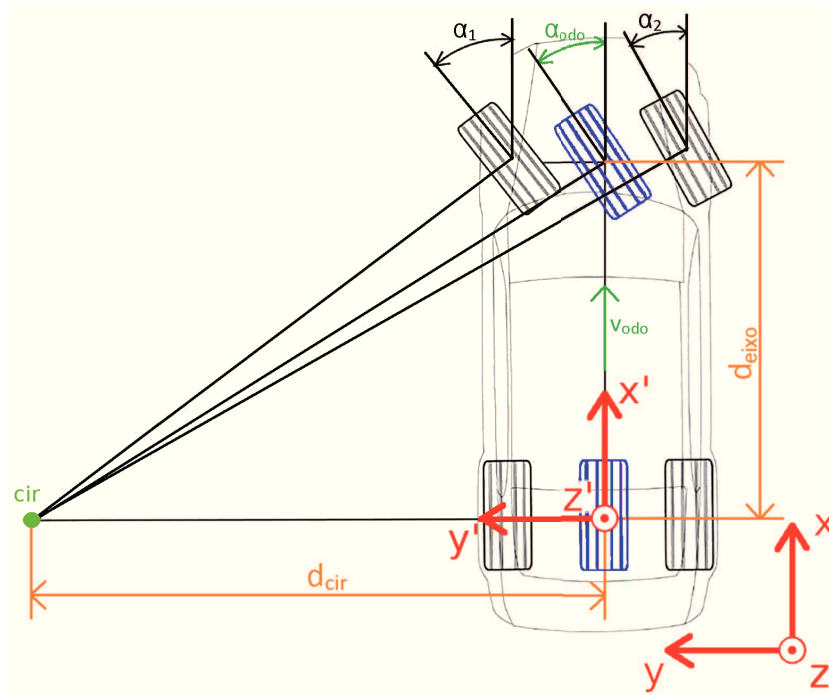


Figura 3.1: Sistema de eixos do veículo.

ângulo da roda para o modelo de um veículo de duas rodas. Através da cinemática direta são retiradas deste modelo as Equações (3.1) e (3.2) correspondentes ao deslocamento angular e linear do veículo [26]:

$$\tan(\alpha_{odo}) = \frac{d_{eixo}}{d_{cir}} \quad (3.1)$$

$$\begin{cases} \Delta\theta = \frac{1}{d_{eixo}} \int_0^t v_{odo}(t) \cdot \tan(\alpha_{odo}) dt \\ \Delta x = \int_0^t v_{odo}(t) \cdot \cos(\theta) dt \\ \Delta y = \int_0^t v_{odo}(t) \cdot \sin(\theta) dt \end{cases} \Leftrightarrow \begin{cases} \Delta\theta = \frac{v_{odo}(t) \cdot \Delta t \cdot \tan(\alpha_{odo})}{d_{eixo}} \\ \Delta x = v_{odo}(t) \cdot \Delta t \cdot \cos(\theta) \\ \Delta y = v_{odo}(t) \cdot \Delta t \cdot \sin(\theta) \end{cases} \quad (3.2)$$

Em que  $v_{odo}$  e  $\alpha_{odo}$  correspondem, respetivamente, à velocidade linear e ângulo das rodas obtidos pelos sensores do veículo;  $d_{eixo}$  à distância entre eixos;  $d_{cir}$  à distância desde o centro da roda virtual traseira ao centro instantâneo de rotação (cir);  $\Delta\theta$ ,  $\Delta x$  e  $\Delta y$  ao deslocamento angular e linear em X e Y, respetivamente, face ao último estado, no sistema de eixos global;  $\Delta t$  ao intervalo de tempo desde a última aquisição. Tendo em conta a alta frequência de aquisição deste sensor, é assumido que o intervalo de tempo  $\Delta t$  é tão pequeno que a velocidade linear e o ângulo das rodas podem ser consideradas constantes entre aquisições.

A Figura 3.2 exemplifica um deslocamento efetuado entre dois instantes de aquisição dos sensores do veículo. Para melhor interpretação, esta foi representada numa escala

exagerada, podendo assim perceber-se os vários valores obtidos entre dois instantes consecutivos.

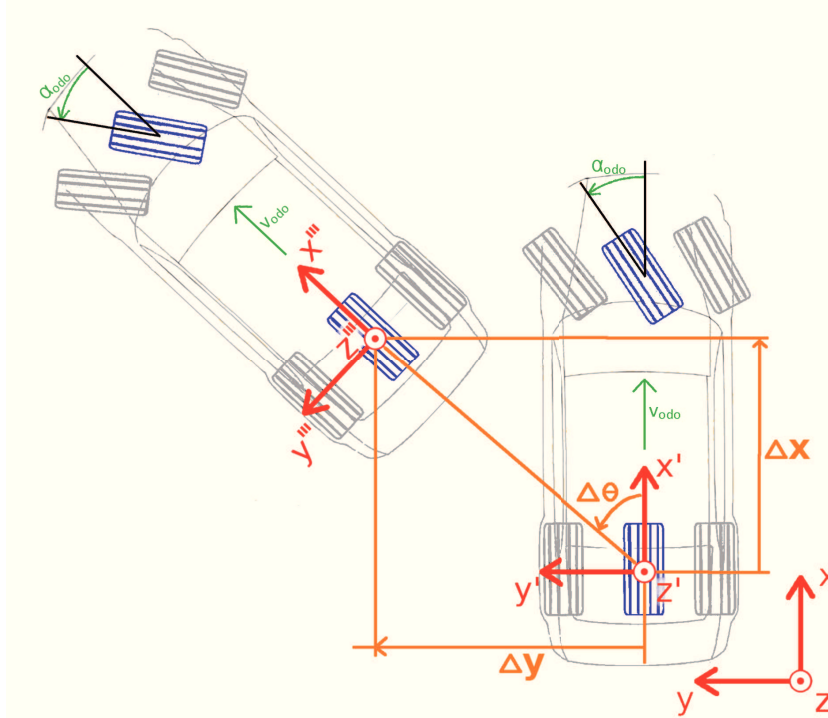


Figura 3.2: Deslocamento de um veículo entre dois instantes em escala exagerada.

### 3.1.2 Sensores inerciais

Como referido anteriormente, a unidade inercial (IMU) permite obter informação das acelerações e das velocidades angulares em três eixos ortogonais. Como o modelo utilizado é 2D, apenas foi considerada a aceleração linear ( $a_{accel}$ ) na direção do movimento do veículo (ao longo de  $X'$ ) e a velocidade angular ( $\omega$ ) de acordo com a rotação feita pelo veículo (em torno de  $Z'$ ).

Através do acelerómetro incorporado na IMU, e utilizando o modelo cinemático para um movimento retilíneo uniformemente acelerado, obtém-se o deslocamento linear ( $\Delta x_{accel}$ ) e a velocidade linear ( $v_{k+1}$ ) desde a última estimativa, ambos ao longo de  $X'$ :

$$\Delta x_{accel} = v_k \Delta t + \frac{a_{accel} \Delta t^2}{2} \quad (3.3)$$

$$v_{k+1} = v_k + a_{accel} \Delta t \quad (3.4)$$

Em que  $v_k$  corresponde à velocidade linear no estado anterior ( $k$ ), ao longo de  $X'$ .

O deslocamento angular é calculado recorrendo ao giroscópio, utilizando a velocidade

angular ( $\omega$ ) para um determinado intervalo de tempo. Assumindo que a rotação é feita a velocidade constante, resulta na Equação (3.5).

$$\Delta\theta_{giro} = \omega\Delta t \quad (3.5)$$

Algumas IMU são também equipadas com magnetômetros que poderão ter um comportamento semelhante ao de uma bússola. Para melhor interpretação, será denominado de bússola à utilização dos magnetômetros com comportamento semelhante ao de uma bússola. No caso da IMU estar equipada com uma bússola, a orientação disponibilizada por esta ( $\theta_{bussola}$ ) será a mesma que a do veículo ( $\theta_{k+1}$ ) no instante  $k + 1$ , assumindo que a IMU está orientada com o veículo, e que o referencial do mundo está alinhado com o referencial magnético da bússola. Na prática este  $\theta_{bussola}$  é calculado em função dos magnetômetros e dos acelerômetros [27]. A Equação (3.6), a Equação (3.7) e a Equação (3.8) refletem o cálculo da orientação do veículo com base na IMU.

$$Y_h = -m_y \cos(Roll) \sin(Pitch) + m_z \sin(Roll) \quad (3.6)$$

$$X_h = m_x \cos(Pitch) + m_y \sin(Pitch) * \sin(Roll) + m_z \sin(Pitch) \cos(Roll) \quad (3.7)$$

$$\theta_{bussola} = atan2(Y_h, X_h) \quad (3.8)$$

Sendo que *Roll* e *Pitch* correspondem à Equação (3.9) e à Equação (3.10), respectivamente;  $m_x$ ,  $m_y$  e  $m_z$  correspondem ao valor medido pelos magnetômetros nos eixos X', Y' e Z', respectivamente,

$$Roll = atan2(a_x, g) \quad (3.9)$$

$$Pitch = atan2(a_y, a_z) \quad (3.10)$$

Onde  $a_x$ ,  $a_y$  e  $a_z$  correspondem à aceleração linear medida pelos acelerômetros no eixo dos X', Y' e Z', respectivamente;  $g$  corresponde à aceleração gravítica.

Os magnetômetros são particularmente sensíveis a fatores externos, nomeadamente à influência de estruturas metálicas, de cabos de alta tensão ou equipamentos elétricos. Para minimizar impactos de possíveis desvios na orientação, foi considerado apenas o deslocamento angular ( $\Delta\theta_{bussola}$ ) ao invés da orientação absoluta do veículo. Sendo assim, o modelo de medidas da IMU é definido pela Equação 3.11.

$$\theta_{k+1} = \theta_k + \Delta\theta_{bussola} \quad (3.11)$$

À semelhança da odometria, os sensores inerciais também têm alta frequência de aquisição, sendo também aqui assumido que o intervalo de tempo  $\Delta t$  é tão pequeno que

a aceleração linear e a velocidade angular sejam constantes entre aquisições.

### 3.1.3 GPS

O GPS transmite-nos a localização absoluta em coordenadas geográficas. Visto o modelo desejado ser em coordenadas cartesianas, será necessário fazer uma conversão de coordenadas.

O denominado sistema de eixos terrestre, tem a origem situada no centro de massa da Terra, em que o eixo  $X''$  é coincidente com o plano equatorial e positivo na direção de longitude  $0^\circ$ , o eixo  $Y''$  também é coincidente ao plano equatorial, mas positivo na direção de longitude  $90^\circ$  e o eixo  $Z''$  é paralelo ao eixo de rotação da Terra e positivo na direção norte. A Figura 3.3 representa a transformação de coordenadas geográficas para cartesianas.

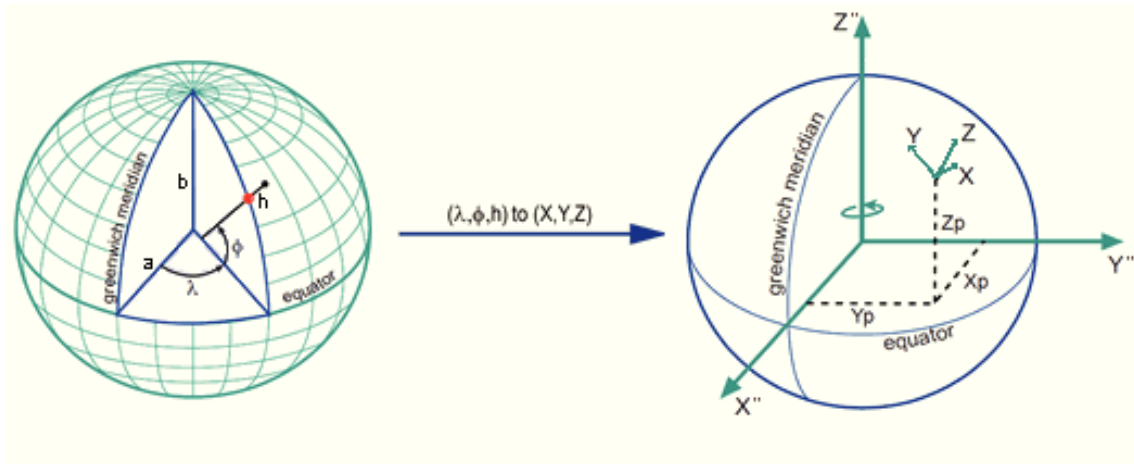


Figura 3.3: Transformação de coordenadas geográficas para cartesianas [28].

A transformação de coordenadas geográficas para cartesianas consiste nas seguintes expressões matemáticas [29], Equação (3.12):

$$\begin{cases} X'' = (N + h) \cos(\varphi) \cos(\lambda) \\ Y'' = (N + h) \cos(\varphi) \sin(\lambda) \\ Z'' = ((b^2/a^2)N + h) \sin(\varphi) \end{cases} \quad (3.12)$$

Onde  $X''$ ,  $Y''$  e  $Z''$  correspondem às coordenadas cartesianas e  $\varphi$ ,  $\lambda$  e  $h$  correspondem à latitude, longitude e à altitude da posição em análise. O raio do semi-eixo maior/equatorial da esferoide terrestre tem o comprimento de  $a = 6378.1370$  km enquanto que o semi-eixo menor/polar de  $b = 6356.7523$  km [30]. A distância do ponto ao centro da Terra é calculada por:  $N = \frac{a^2}{\sqrt{a^2 \cos^2(\varphi) + b^2 \sin^2(\varphi)}}$ .

De modo a conciliar os valores obtidos pela odometria e pela IMU com os valores obtidos pelo GPS, é feita uma transformação de coordenadas, visto terem como referência um sistema de eixos diferentes (sistema de eixos global X,Y e Z para a odometria e a IMU e sistema de eixos terrestre X'', Y'' e Z'' para o GPS). A opção de realizar a transformação de coordenadas para o sistema de eixos global ao invés de para o sistema de eixos terrestre, deve-se ao facto da frequência de aquisição do GPS ser bastante inferior à da aquisição da IMU em conjunto com a odometria, minimizando assim a utilização de recursos no processamento de dados.

A transformação de coordenadas terrestres para o sistema de eixos global consiste na aplicação das matrizes de rotação e de translação de coordenadas. Na Figura 3.4 estão representadas as rotações e translações necessárias por ordem de realização. Verifica-se então que a primeira rotação a realizar será em torno de Z'', a segunda em torno de Y'', de novo em torno de Z'' e, por fim, translação em X'', Y'' e Z''.

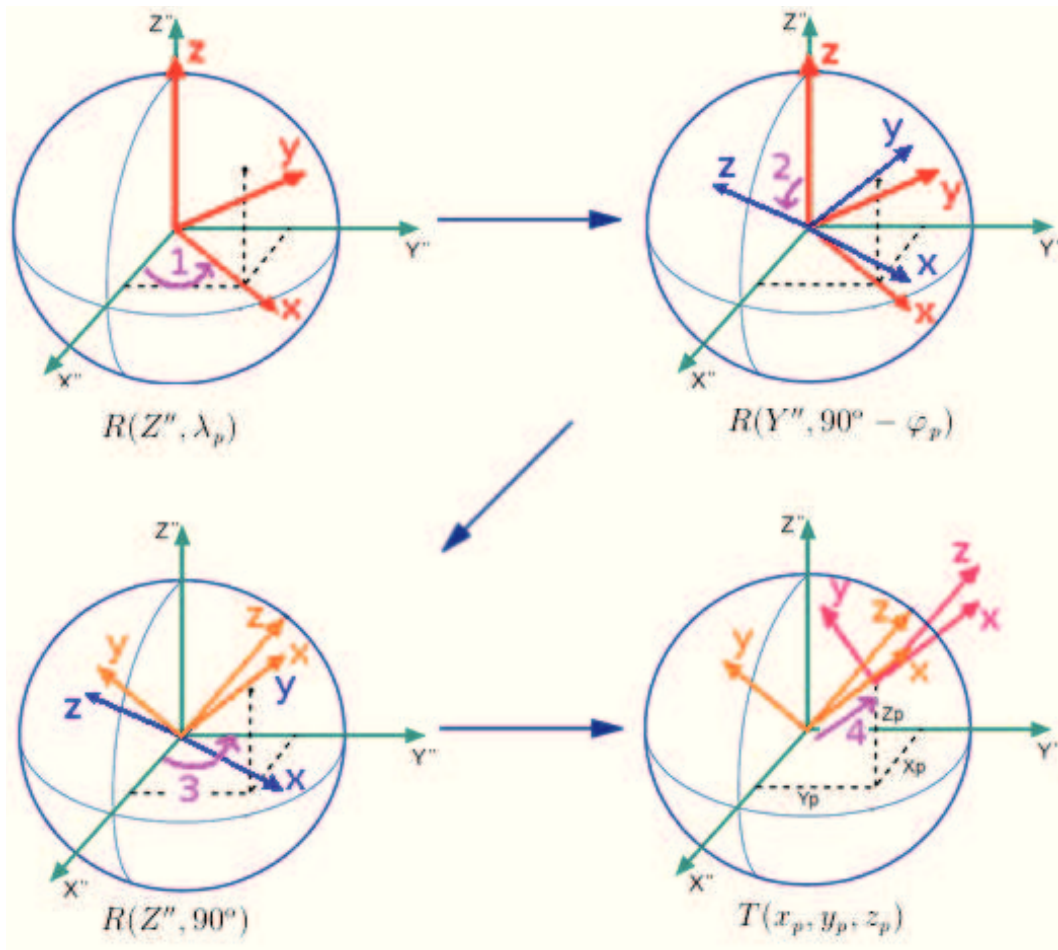


Figura 3.4: Rotação de sistema de eixos.

A Equação (3.13) representa as matrizes genéricas de rotação em torno de Y'' ( $R(Y'', \beta)$ ) e de Z'' ( $R(Z'', \gamma)$ ) enquanto que a Equação (3.14) representa a matriz genérica de translação ( $T(x, y, z)$ ) [31].

$$R(Y'', \beta) = \begin{bmatrix} \cos(\beta) & 0 & -\sin(\beta) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\beta) & 0 & \cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} R(Z'', \gamma) = \begin{bmatrix} \cos(\gamma) & \sin(\gamma) & 0 & 0 \\ -\sin(\gamma) & \cos(\gamma) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.13)$$

$$T(x, y, z) = \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.14)$$

Em que  $\beta$  e  $\gamma$  correspondem ao ângulo de rotação em torno de  $Y''$  e  $Z''$ , respetivamente, e  $x$ ,  $y$  e  $z$  correspondem a translação efetuada em  $X''$ ,  $Y''$  e  $Z''$ , respetivamente. Estes ângulos de rotação são retirados através da Figura 3.4, sendo o ângulo da primeira rotação  $\gamma = \lambda_p$ , o ângulo da segunda rotação  $\beta = 90^\circ - \varphi_p$  e o último ângulo de  $\gamma = 90^\circ$ . As medidas de translação também podem ser retiradas da mesma figura, sendo a translação em  $X''$  de  $x = x_p$ , em  $Y''$  de  $y = y_p$  e em  $Z''$  de  $z = z_p$ . Onde  $\varphi_p$  e  $\lambda_p$  correspondem à latitude e longitude da posição inicial, respetivamente;  $x_p$ ,  $y_p$  e  $z_p$  correspondem ao valor em  $X''$ ,  $Y''$ , e  $Z''$  da posição inicial, respetivamente. Substituindo os ângulos e as translações nas matrizes genéricas, obtém-se o seguinte resultado:

$$R(Z'', \lambda_p) = \begin{bmatrix} \cos(\lambda) & \sin(\lambda) & 0 & 0 \\ -\sin(\lambda) & \cos(\lambda) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.15)$$

$$R(Y'', 90^\circ - \varphi_p) = \begin{bmatrix} \cos(90^\circ - \varphi) & 0 & -\sin(90^\circ - \varphi) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(90^\circ - \varphi) & 0 & \cos(90^\circ - \varphi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.16)$$

$$R(Z'', 90^\circ) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.17)$$

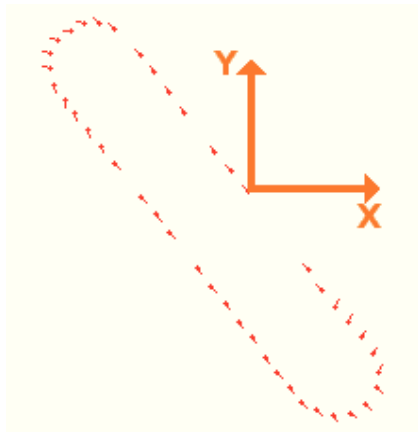
$$T(-x_p, -y_p, -z_p) = \begin{bmatrix} 1 & 0 & 0 & -x_p \\ 0 & 1 & 0 & -y_p \\ 0 & 0 & 1 & -z_p \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.18)$$

Como este modelo terá como princípio o funcionamento apenas no plano tangente à superfície da Terra, a altitude (medida referente ao eixo Z do sistema global) pode ser descartada, passando assim de um modelo 3D para um modelo 2D. Aplicando estas equações, são transformadas as coordenadas referentes aos sistema de eixos terrestre (Equação (3.12)) em coordenadas referentes ao sistema de eixos global, ver Equação 3.19.

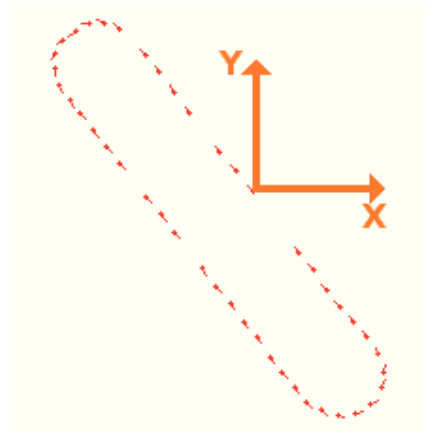
$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = R(Z'', 90^\circ)R(Y'', 90^\circ - \varphi_p)R(Z'', \lambda_p)T(-x_p, -y_p, 0) \begin{bmatrix} X'' \\ Y'' \\ Z'' \end{bmatrix} \quad (3.19)$$

A orientação disponibilizada pelo GPS ( $\theta_{gps}$ ) é estimada, pelo sensor, em função de um determinado número de posições adquiridas. Verificou-se em testes experimentais que, para maiores deslocamentos angulares, a orientação fornecida pelo GPS sofre de maior erro de atualização quanto maior for o número de aquisições utilizadas no cálculo. Na Figura 3.5 é apresentado o percurso medido pelo GPS, para um teste experimental efetuado em ambiente real, cuja finalidade é a comparação entre o  $\theta_{gps}$  medido pelo GPS (ver Figura 3.5(a)) e o  $\theta_{gps}$  estimado em função das duas últimas aquisições (ver Figura 3.5(b)). É então possível observar que a orientação medida pelo GPS, tem um atraso bastante significativo. Não estando disponível a configuração do número de posições utilizadas na estimação da orientação pelo sensor, o  $\theta_{gps}$  foi estimado externamente com base na posição atual e na posição anterior, ver Equação 3.20.

$$\theta_{gps} = atan2(y_{k+1} - y_k, x_{k+1} - x_k) \quad (3.20)$$



(a)  $\theta_{gps}$  medido do GPS.



(b) Com correção de  $\theta_{gps}$ .

Figura 3.5: Percurso medido do GPS, com e sem correção da orientação.

## 3.2 Método baseado no Filtro Estendido de Kalman

Na definição original do EKF a informação sensorial é apenas utilizada no passo de observação. No entanto é comum, tal como proposto em [22], utilizar a odometria no passo de predição, não obstante corresponder de facto a informação sensorial. Desta forma considera-se a odometria como sinais de controlo, assumindo-se assim que esta fornece uma melhor estimativa na dinâmica do sistema do que os sinais de controlo. Como tal, a fim de comparar resultados, foram implementadas as duas formas para o mesmo sistema. Em ambos os métodos, é feita a fusão sensorial entre a odometria ( $v_{odo}$  e  $\alpha_{odo}$ ), os sensores inerciais ( $\theta_{bussola}$ ,  $\omega_{giro}$  e  $a_{acel}$ ) e o GPS ( $x_{gps}$ ,  $y_{gps}$ ,  $\theta_{gps}$  e  $v_{gps}$ ), tendo como única diferença o processo onde a odometria é considerada. Os sinais de controlo não foram tidos em consideração para avaliar o impacto no sistema, estando deste modo o sistema preparado para utilização em condução autónoma ou manual. Assim, na ausência de sinais de controlo, é assumido que a dinâmica de estado se mantém. A Figura 3.21 representa o modelo genérico para este sistema.

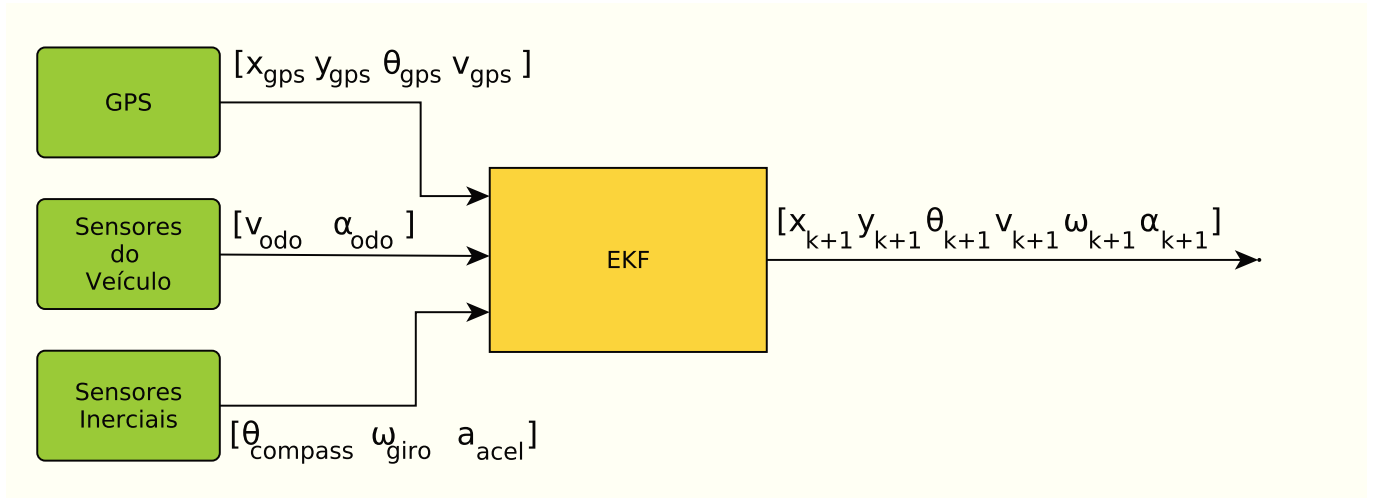


Figura 3.6: Diagrama para o processo de fusão sensorial.

Sendo o sistema diferenciado apenas no processo onde a odometria é considerada, o vetor de estado é comum para os dois métodos, sendo definido pela Equação 3.21.

$$X_{(k+1)} = [x_{(k+1)} \quad y_{(k+1)} \quad \theta_{(k+1)} \quad v_{(k+1)} \quad \omega_{(k+1)} \quad a_{(k+1)}]^T \quad (3.21)$$

### 3.2.1 EKF com odometria no passo de correção

Neste caso aplica-se o filtro na sua forma clássica, isto é, utilizando a odometria como um sensor, ou seja, são utilizados todos os sensores no processo de atualização, sendo tipicamente a predição baseada no estado anterior e nos sinais de controlo, doravante este

filtro será denominado EKF<sub>1</sub>. Como neste caso os sinais de controlo não foram tidos em consideração, a predição é baseada apenas no estado anterior.

## Predição

Como referido anteriormente, neste modelo a predição é feita sem recorrer a qualquer tipo de sinal de controlo, baseando-se apenas do estado anterior e na dinâmica de estado, tendo assim um modelo puramente preditivo. Derivado da Equação (2.3), o modelo de estimação do estado é definido pela Equação (3.22) e tem como matrizes de jacobiano do sistema em função do estado e em função da entrada a Equação (3.23) e a Equação (3.24), respetivamente. Como é assumido que a dinâmica do estado se mantém, a aceleração linear ( $a_k$ ) e a velocidade angular ( $\omega_k$ ) são consideradas constantes.

$$f = \begin{cases} x_{k+1} = x_k + \Delta x \\ y_{k+1} = y_k + \Delta y \\ \theta_{k+1} = \theta_k + \Delta\theta \\ v_{k+1} = v + \Delta v \\ \omega_{k+1} = \omega \\ a_{k+1} = a \end{cases} \Leftrightarrow f = \begin{cases} x_{k+1} = x_k + (v_k \cdot \Delta t + \frac{a_k \cdot \Delta t^2}{2}) \cos(\theta_k) \\ y_{k+1} = y_k + (v_k \cdot \Delta t + \frac{a_k \cdot \Delta t^2}{2}) \sin(\theta_k) \\ \theta_{k+1} = \theta_k + \omega_k \cdot \Delta t \\ v_{k+1} = v_k + a_k \cdot \Delta t \\ \omega_{k+1} = \omega_k \\ a_{k+1} = a_k \end{cases} \quad (3.22)$$

$$A_{k+1} = \left. \frac{\partial f}{\partial X} \right|_{\substack{X=\hat{X}_k \\ u=\hat{u}_k}} = \begin{bmatrix} 1 & 0 & -(v_k \cdot \Delta t + \frac{a_k \cdot \Delta t^2}{2}) \sin(\theta_k) & \Delta t \cdot \cos(\theta_k) & 0 & \frac{\Delta t^2}{2} \cos(\theta_k) \\ 0 & 1 & (v_k \cdot \Delta t + \frac{a_k \cdot \Delta t^2}{2}) \cos(\theta_k) & \Delta t \cdot \sin(\theta_k) & 0 & \frac{\Delta t^2}{2} \sin(\theta_k) \\ 0 & 0 & 1 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.23)$$

$$B_{k+1} = \left. \frac{\partial f}{\partial u} \right|_{\substack{X=\hat{X}_k \\ u=\hat{u}_k}} = [0] \quad (3.24)$$

Onde  $\Delta t$  corresponde ao intervalo de tempo desde a última predição e  $[0]$  corresponde a uma matriz nula. Posteriormente, é atualizada a covariância da estimativa do estado através da Equação (2.4). Terminado este passo, é feita então a correção.

## Correção

Considerando os três sensores no processo de correção, resulta que o vetor de medidas é composto pelas medidas de todos os sensores, sendo eles a velocidade linear e ângulo das rodas obtidas através da odometria ( $v_{odo}$  e  $\alpha_{odo}$ ), a aceleração linear e angular obtida pelos sensores inerciais ( $\theta_{bussola}$ ,  $\omega_{giro}$  e  $a_{accel}$ ) e a posição e orientação absolutas e velocidade linear obtidas do GPS ( $x_{gps}$ ,  $y_{gps}$ ,  $\theta_{gps}$ ,  $v_{gps}$ ) - Equação (3.25).

$$z_{k+1} = \begin{bmatrix} v_{odo} \\ \alpha_{odo} \\ \theta_{bussola} \\ \omega_{giro} \\ a_{accel} \\ x_{gps} \\ y_{gps} \\ \theta_{gps} \\ v_{gps} \\ \frac{\theta_{gps} - last\_ \theta_{gps}}{\Delta t_{gps}} \\ \frac{v_{gps} - last\_ v_{gps}}{\Delta t_{gps}} \end{bmatrix} \quad (3.25)$$

Onde  $\theta_{last\_gps}$  e  $v_{last\_gps}$  correspondem à orientação e à velocidade linear adquiridas pelo GPS no instante anterior;  $\Delta t_{gps}$  ao intervalo de tempo desde a última aquisição de dados do GPS.

Derivado da Equação (2.2), considerando o ruído nulo de modo a obter a melhor estimativa, resulta no modelo de medidas estimadas da Equação (3.26)).

$$\hat{z}_{k+1} = h(\hat{X}_{k+1}^-) = \begin{bmatrix} \hat{v}_k \\ atan2(\hat{\omega}_k \cdot d_{eixo}, \hat{v}_k) \\ \hat{\theta}_k \\ \hat{\omega}_k \\ \hat{a}_k \\ \hat{x}_k \\ \hat{y}_k \\ \hat{\theta}_k \\ \hat{v}_k \\ \hat{\omega}_k \\ \hat{a}_k \end{bmatrix} \quad (3.26)$$

A matriz do jacobiano das medidas, em função do estado, resulta na Equação (3.27):

$$H_{k+1} = \left. \frac{\partial h}{\partial X} \right|_{X=\hat{X}_{k+1}^-} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{d_{eixo} \cdot \hat{\omega}_k \cdot \hat{v}_k \cdot \sqrt{\hat{v}_k}}{\hat{v}_k^2 + (\hat{\omega}_k \cdot d_{eixo})^2} & \frac{d_{eixo} \cdot \hat{v}_k}{\hat{v}_k^2 + (\hat{\omega}_k \cdot d_{eixo})^2} & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.27)$$

### 3.2.2 EKF com odometria no passo de predição

Tal como em [22], dado que a odometria devolve informação interna relativamente ao deslocamento do robô, esta pode ser considerada como uma predição do movimento realizado entre estados consecutivos, podendo ser utilizada no filtro como sinal de controlo ( $u_k$ ), de agora em diante este filtro será denominado EKF\_2. Neste caso, o vetor de entradas é definido pela Equação (3.28).

$$u_{k+1} = [v_{odo} \quad \alpha_{odo}]^T \quad (3.28)$$

#### Predição

Derivado da Equação (2.3), no processo de predição através da odometria, o modelo de estimação de estado é definido pela Equação (3.29) e tem como matrizes de jacobiano do sistema em função do estado estimado ( $\hat{X}_k$ ) e em função da entrada ( $u_k$ ) a Equação (3.30) e (3.31), respetivamente. De novo, é também atualizada a covariância da estimativa do estado da através Equação (2.4).

$$f = \begin{cases} x_{k+1} = x_k + \Delta x \\ y_{k+1} = y_k + \Delta y \\ \theta_{k+1} = \theta_k + \Delta \theta \\ v_{k+1} = v \\ \omega_{k+1} = \omega \\ a_{k+1} = a \end{cases} \Leftrightarrow f = \begin{cases} x_{k+1} = x_k + (v_k \cdot \Delta t + \frac{a_k \cdot \Delta t^2}{2}) \cos(\theta_k) \\ y_{k+1} = y_k + (v_k \cdot \Delta t + \frac{a_k \cdot \Delta t^2}{2}) \sin(\theta_k) \\ \theta_{k+1} = \theta_k + \omega_k \cdot \Delta t \\ v_{k+1} = v_{odo} \\ \omega_{k+1} = \frac{v_{odo} \cdot \tan(\alpha_{odo})}{d_{eixo}} \\ a_{k+1} = \frac{v_{odo} - v_k}{\Delta t} \end{cases} \quad (3.29)$$

$$A_{k+1} = \frac{\partial f}{\partial X} \Bigg|_{\substack{X=\hat{X}_k \\ u=\hat{u}_k}} = \begin{bmatrix} 1 & 0 & -(v_k \cdot \Delta t + \frac{a_k \cdot \Delta t^2}{2}) \sin(\theta_k) & \Delta t \cdot \cos(\theta_k) & 0 & \frac{\Delta t^2}{2} \cos(\theta_k) \\ 0 & 1 & (v_k \cdot \Delta t + \frac{a_k \cdot \Delta t^2}{2}) \cos(\theta_k) & \Delta t \cdot \sin(\theta_k) & 0 & \frac{\Delta t^2}{2} \sin(\theta_k) \\ 0 & 0 & 1 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{-1}{\Delta t} & 0 & 0 \end{bmatrix} \quad (3.30)$$

$$B_{k+1} = \frac{\partial f}{\partial u} \Bigg|_{\substack{X=\hat{X}_k \\ u=\hat{u}_k}} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ \frac{\tan(\alpha_{odo})}{d_{eixo}} & \frac{v_{odo}}{\cos^2(\alpha_{odo}) \cdot d_{eixo}} \\ \frac{1}{\Delta t} & 0 \end{bmatrix} \quad (3.31)$$

Como esperado, em contraste com o EKF com a odometria no passo de correção, a matriz do jacobiano da dinâmica do sistema em ordem a  $u$ , matriz  $B_{k+1}$ , não é nula.

## Correção

Sendo a odometria utilizada no passo de predição, apenas a IMU e o GPS são utilizados no passo de correção, pelo que a Equação (2.2) resulta no modelo do sensor, dado pelo vetor de medidas (Equação (3.32)) e pelo vetor de medidas esperadas (Equação (3.33)).

$$z_{k+1} = \begin{bmatrix} \theta_{bussola} \\ \omega_{giro} \\ a_{acel} \\ x_{gps} \\ y_{gps} \\ \theta_{gps} \\ v_{gps} \\ \frac{\theta_{gps} - last\_ \theta_{gps}}{\Delta t_{gps}} \\ \frac{v_{gps} - last\_ v_{gps}}{\Delta t_{gps}} \end{bmatrix} \quad (3.32)$$

$$\hat{z}_{k+1} = h(\hat{X}_{k+1}^-) = \begin{bmatrix} \hat{\theta}_k \\ \hat{\omega}_k \\ \hat{a}_k \\ \hat{x}_k \\ \hat{y}_k \\ \hat{\theta}_k \\ \hat{v}_k \\ \hat{\omega}_k \\ \hat{a}_k \end{bmatrix} \quad (3.33)$$

Assim, a matriz do jacobiano do modelo das medidas (Equação (2.10)) resulta na Equação (3.34):

$$H_{k+1} = \left. \frac{\partial h}{\partial X} \right|_{X=\hat{X}_{k+1}^-} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.34)$$

### 3.3 Método baseado no Filtro de Partículas

Prosseguindo com a implementação para o mesmo sistema, também os sensores utilizados são os mesmos: odometria ( $v_{odo}$  e  $\alpha_{odo}$ ), sensores inerciais ( $\theta_{bussola}$ ,  $\omega_{giro}$  e  $a_{acel}$ ) e GPS ( $x_{gps}$ ,  $y_{gps}$ ,  $\theta_{gps}$  e  $v_{gps}$ ). Tal como no filtro EKF, inicialmente é definido o vetor de estado de acordo com o modelo de sistema (Equação (2.11)). A Equação (3.35) representa o vetor do estado.

$$X_{(k+1)} = \begin{bmatrix} x_{(k+1)} & y_{(k+1)} & \theta_{(k+1)} & v_{(k+1)} & \omega_{(k+1)} & a_{(k+1)} \end{bmatrix}^T \quad (3.35)$$

#### 3.3.1 Predição

A deslocação das partículas em função da dinâmica do estado, é realizada individualmente tendo por base o modelo de estimação de estado, ao qual é adicionado um ruído aleatório de acordo com o modelo da dinâmica de estado. A Equação (3.36) representa o modelo

do sistema para a predição com base na odometria.

$$f = \begin{cases} x_{k+1} = x_k + \Delta x \\ y_{k+1} = y_k + \Delta y \\ \theta_{k+1} = \theta_k + \Delta\theta \\ v_{k+1} = v \\ \omega_{k+1} = \omega \\ a_{k+1} = a \end{cases} \Leftrightarrow f = \begin{cases} x_{k+1} = x_k + (v_k \cdot \Delta t + \frac{a_k \cdot \Delta t^2}{2}) \cos(\theta_k) \\ y_{k+1} = y_k + (v_k \cdot \Delta t + \frac{a_k \cdot \Delta t^2}{2}) \sin(\theta_k) \\ \theta_{k+1} = \theta_k + \omega_k \cdot \Delta t \\ v_{k+1} = v_{odo} \\ \omega_{k+1} = \frac{v_{odo} \cdot \tan(\alpha_{odo})}{\text{deixo}} \\ a_{k+1} = \frac{v_{odo} - v_k}{\Delta t} \end{cases} \quad (3.36)$$

### 3.3.2 Atualização

Definido o vetor de medidas (Equação (3.37)) e o vetor de medidas estimadas (Equação (3.38)), derivado do modelo dos sensores (Equação (2.12)), é feita a observação, comparando individualmente as medidas lidas com as estimadas de cada partícula. Em função desta comparação, é calculado o peso de cada partícula - Equação (2.14). Após a atualização dos pesos, é feita a sua normalização através da Equação (2.15).

$$z_{k+1} = \begin{bmatrix} \theta_{bussola} \\ \omega_{giro} \\ a_{acel} \\ x_{gps} \\ y_{gps} \\ \theta_{gps} \\ v_{gps} \\ \frac{\theta_{gps-last\_}\theta_{gps}}{\Delta t_{gps}} \\ \frac{v_{gps-last\_}v_{gps}}{\Delta t_{gps}} \end{bmatrix} \quad (3.37)$$

$$\hat{z}_{k+1} = h(\hat{X}_{k+1}^-) = \begin{bmatrix} \hat{\theta}_k \\ \hat{\omega}_k \\ \hat{a}_k \\ \hat{x}_k \\ \hat{y}_k \\ \hat{\theta}_k \\ \hat{v}_k \\ \hat{\omega}_k \\ \hat{a}_k \end{bmatrix} \quad (3.38)$$

### 3.3.3 Reamostragem

A reamostragem consiste num modelo genérico que replica as partículas com maior probabilidade proporcional ao seu peso de modo a obter uma densidade da nuvem de partículas mais densa em torno das partículas de maior peso. Assim sendo, este passo é efetuado de acordo com o que foi explicado na Subsecção 2.3.3.



# Capítulo 4

## Implementação prática

Neste capítulo são apresentados os ambientes utilizados e a implementação realizada para pôr em prática os métodos descritos no Capítulo 3.

### 4.1 Sistema base

O sistema base utilizado consistiu no *software Robot Operating System* (ROS) [32], que é constituído por ferramentas e bibliotecas que simplificam e tornam a programação de robôs mais portátil, independente de hardware, permitindo também um aumento mais controlado da complexidade do sistema implementado. Uma das suas funcionalidades é a criação de nós para comunicação entre sistemas. A comunicação entre nós é feita através de *streaming* de tópicos e serve para, por exemplo, fazer a comunicação entre um robô e uma aplicação. Uma das vantagens desta utilização é de ter um sistema "ligar e usar" (do inglês: *plug and play*), sendo assim possível, por exemplo, comutar entre a simulação e o veículo real sem alterar a aplicação que implementa o sistema de localização (ver Figura 4.1).

A aplicação de localização desenvolvida, denominada SLVA (Sistema de Localização de um Veículo Automóvel), é composta apenas por um nó e vários tópicos, um por cada sensor e por cada sinal de controlo, que são utilizados para fazer a comunicação entre o sistema e o veículo real, ou simulador, dependendo se está a ser utilizado o ambiente real ou o ambiente de simulação, respetivamente. Assim, a comutação entre ambientes é automática, bastando arrancar com o simulador ou com o nó que comunica com o veículo real, dependendo do pretendido.

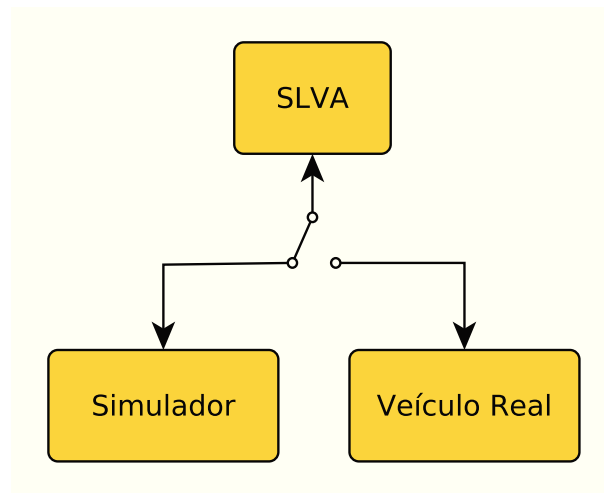


Figura 4.1: Diagrama de comutação da aplicação entre ambiente de simulação e real.

## 4.2 Ambiente de Simulação

Com intuito de minimizar a utilização do veículo real, numa primeira fase foram realizados testes em ambiente de simulação. Esta tem a vantagem de não ter restrições nem custos de utilização, evitando despesas desnecessárias. O *software* utilizado para simulação foi o Gazebo [33], o qual consiste num simulador gratuito, de código aberto, capaz de simular vários modelos realistas de robôs, sensores e objetos, tendo em conta diversas características físicas, tais como o peso, o atrito e a inércia.

O modelo de veículo utilizado no ambiente de simulação é baseado num Polaris Ranger EV, desenvolvido para as provas do *DARPA Robotics Challenge*<sup>1</sup>, modelo este bastante realista, tanto em aspetos visuais como no desempenho (ver Figura 4.2). Com um comportamento semelhante a um automóvel real, o controlo do veículo é feito através do ângulo do pedal do acelerador, do pedal do travão e do ângulo do volante. De modo a ter um controlo do modelo semelhante ao presente no veículo real utilizado, detalhado na Secção 4.3, foram implementados controladores ao nível do Gazebo, que controlam o veículo virtual através da velocidade linear e do ângulo da direção pretendidos, os quais se descrevem de seguida.

A Figura 4.3 representa os tópicos de comunicação utilizados para comunicação entre o simulador e o sistema. A aplicação desenvolvida é composta apenas por um nó (SLVA) e vários tópicos, um por cada sensor e por cada sinal de controlo, que são utilizados para fazer a comunicação entre o Gazebo e o sistema, no ambiente de simulação. Adicionalmente, foi também adicionado o estado estimado pelo SLVA, para efeitos de controlo, e o estado real, com o propósito de realização de *debug* ao sistema.

<sup>1</sup><http://gazebosim.org/wiki/DRC>



Figura 4.2: *Polaris Ranger EV - veículo utilizado para testes em simulação.*

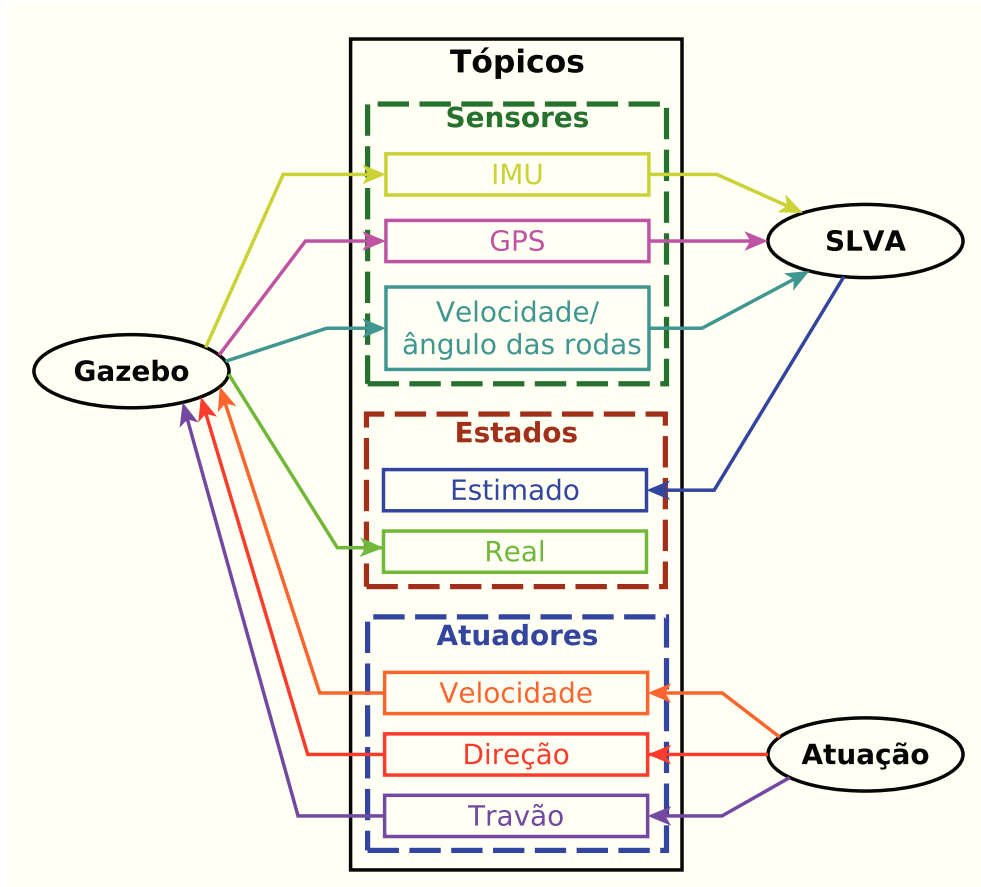


Figura 4.3: *Tópicos de comunicação utilizados em ambiente de simulação.*

## Controlo

Para efeitos de controlo, foi dimensionado um controlador de velocidade e um de direção, resultando assim num modo de atuação semelhante ao veículo real. O controlo da velocidade foi realizado através do controlo do ângulo do pedal do acelerador. Variando este ângulo, obtêm-se diferentes acelerações, tal como mostra a Figura 4.4, onde é representada a evolução da velocidade linear, ao longo do tempo para diferentes ângulos do acelerador.

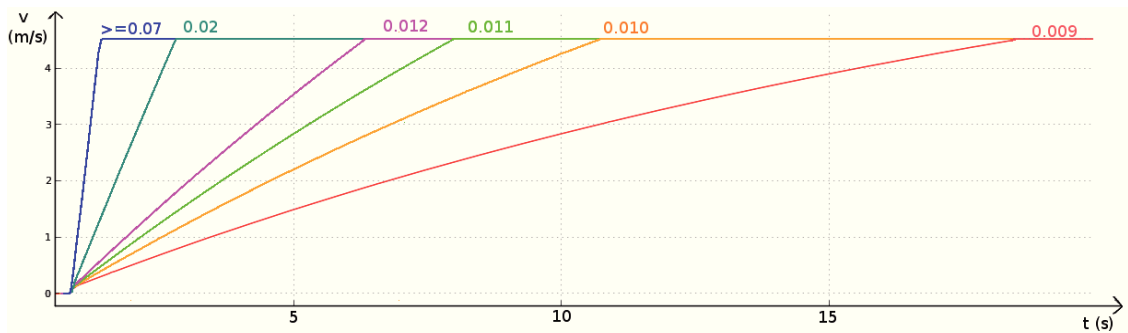


Figura 4.4: *Evolução de velocidade linear para diferentes ângulos do acelerador.*

Onde os números, em cima de cada linha, correspondem ao ângulo do pedal do acelerador, numa escala de  $[0;1]$ . Na Figura 4.4 é possível concluir que, independentemente do ângulo do acelerador, o modelo do veículo tem uma velocidade linear limitada a 4.5 m/s. É também possível retirar que a aceleração linear é aproximadamente constante, para o mesmo ângulo do acelerador, e ainda a gama de funcionamento do pedal do acelerador, sendo ela compreendida entre  $[0, 0.07]$ .

Definidas as circunstâncias de funcionamento, foi implementado um controlador do tipo PI ao nível do Gazebo, simulando assim um controlador de velocidade linear de baixo-nível do veículo simulado. Na Figura 4.5 está representado o esquema de um controlador do tipo Proporcional Integral (PI) [34].

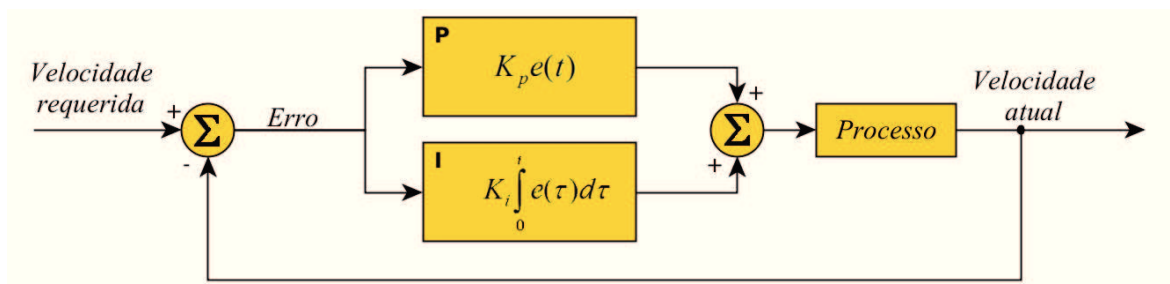


Figura 4.5: *Controlador do tipo PI.*

O *Erro* consiste na diferença entre a velocidade pretendida e a atual; calculados empiricamente,  $K_p$  e  $K_i$  são os ganhos relativo ao controlador proporcional e integral,

respetivamente; o *Processo* consiste no modelo de simulação do veículo, o qual tem como entrada o acelerador e como resposta/saída a velocidade atual;  $t$  corresponde ao intervalo de tempo decorrido desde a instância.

O controlo do ângulo das rodas é feito através da rotação do volante, como tal, para obter o ângulo do volante correspondente ao ângulo das rodas desejado, é preciso aplicar o fator escala do ângulo das rodas com o ângulo de rotação do volante. O ângulo de rotação do volante necessário para obter o ângulo nas rodas pretendido, é dado pela Equação 4.1.

$$\alpha_{\text{volante}} = \alpha_{\text{rodas}} * \frac{\alpha_{\text{max\_volante}}}{\alpha_{\text{max\_rodas}}} \quad (4.1)$$

Onde  $\alpha_{\text{volante}}$  diz respeito ao ângulo do volante correspondente ao ângulo pretendido para as rodas;  $\alpha_{\text{rodas}}$  ao ângulo pretendido para as rodas; o  $\frac{\alpha_{\text{max\_volante}}}{\alpha_{\text{max\_rodas}}}$  corresponde ao fator de escala, calculado pela divisão do ângulo máximo efetuado pelo volante pelo respetivo ângulo das rodas. Neste caso não é necessário implementar nenhum sistema para controlo, visto que a orientação do volante é diretamente definida consoante o sinal de controlo recebido.

O controlo do travão foi implementado em modo binário (ligado/desligado) pois, para as velocidades pretendidas, a simples desaceleração do veículo é suficiente para imobilizar o veículo virtual. Ainda assim, foi implementado o controlo do mesmo para poder manter o veículo em repouso, ou ainda para situações de emergência e/ou estacionamento.

## Sensores

Foram implementados os três sensores utilizados neste trabalho: odometria, GPS e IMU. A implementação da odometria consistiu na determinação no ambiente de simulação da velocidade linear do veículo e da orientação das rodas. No caso do GPS, a implementação consistiu em obter, no ambiente de simulação, a posição absoluta em coordenadas cartesianas, convertidas para coordenadas geográficas através da Equação 4.2 [29] que, basicamente, consiste no inverso da Equação (3.12).

$$\begin{cases} \varphi = \arctan\left(\frac{Z'' + e'^2 b \sin^3(\theta)}{p - e'^2 a \cos^3(\theta)}\right) \\ \lambda = \arctan\left(\frac{Y''}{X''}\right) \\ h = \frac{p}{\cos(\varphi)} - N \end{cases} \quad (4.2)$$

Recordando,  $X'', Y''$  e  $Z''$  correspondem às coordenadas cartesianas e  $\varphi$ ,  $\lambda$  e  $h$  correspondem à latitude, longitude e à altitude da posição em análise. O raio do semi-eixo maior/equatorial da esferoide terrestre tem o comprimento de  $a = 6378.1370$  km enquanto que o semi-eixo menor/polar de  $b = 6356.7523$  km [30]. A distância do ponto ao centro

da Terra é calculada por:  $N = \frac{a^2}{\sqrt{a^2 \cos^2(\varphi) + b^2 \sin^2(\varphi)}}$ .

A simulação dos sensores inerciais já está disponível no Gazebo [33], tendo sido adaptada, com alterações mínimas, para a aplicação em causa, sendo constituída pelas acelerações lineares e velocidades angulares nos três eixos ortogonais e as orientações nos ângulos de Euler (Roll, Pitch e Yaw).

De forma a ter uma simulação mais realista, foi adicionado ruído gaussiano e configurada uma frequência de aquisição na mesma ordem de grandeza da do veículo real e sensores reais (ver Secção 4.3). Assim sendo, as frequências de atualização definidas foram:

- **GPS:** 1 Hz
- **IMU:** 100 Hz
- **Odometria:** 1200 Hz

### 4.3 Ambiente Real

A implementação em ambiente real teve por base o projeto [14], o qual consistiu na implementação de um sistema de controlo de baixo nível de um Volkswagen Sharan vr6 (Figura 4.6) para executar diretivas de condução autónoma. Este veículo é constituído por acionamentos de controlo, nomeadamente o acelerador, o travão, a direção e a caixa de velocidades, e por sensores, sendo eles codificadores nas rodas e na coluna de direção.



Figura 4.6: *Volkswagen Sharan vr6*.

Para além do *hardware*, o *software* implementado permite controlar a velocidade, a direção, o travão (apenas ligado/desligado) e a caixa de velocidades, assim como a comunicação com o veículo. Esta funcionalidade é implementada através do módulo de controlo de baixo nível (MCBN), ilustrado na Figura 4.7.

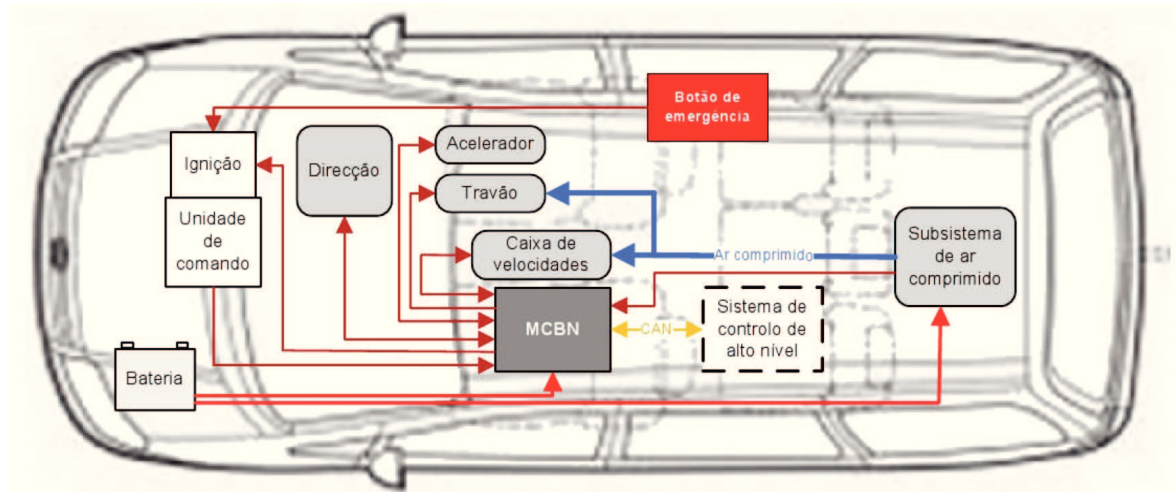


Figura 4.7: *Estrutura do sistema desenvolvido na dissertação anterior [14].*

A comunicação entre o MCBN e o sistema de alto-nível, também desenvolvida no projeto anterior [14], é realizada através de comunicação CAN (ver Apêndice A.1.1).

### Sensores existentes

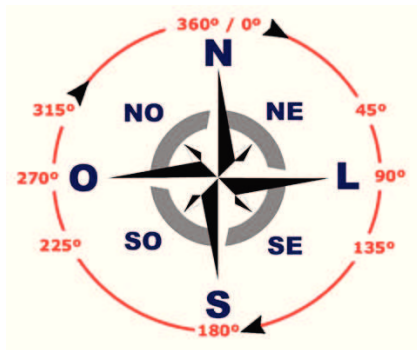
Como referido anteriormente, fruto do projeto anterior, o veículo já está equipado com alguns sensores, sendo eles codificadores nas rodas, originalmente pertencentes ao sistema de ABS, e codificadores na coluna de direção. Através do cálculo da cinemática direta no MCBN, é disponibilizada a velocidade linear do veículo e ângulo absoluto das rodas, posteriormente utilizados para calcular a odometria. No Apêndice A.2.2 é descrita a calibração efetuada para o bom funcionamento deste sensor.

### Sensores adicionados

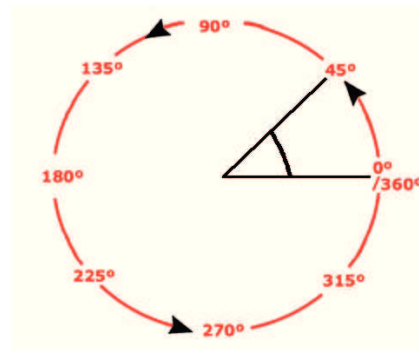
Os sensores não disponíveis anteriormente, e por isso adicionados ao veículo, foram a IMU e o GPS. A IMU utilizada tem a referência MinIMU-9 v2 (Figura. A.4), e é constituída por acelerómetros, giroscópios e magnetómetros, todos em três eixos ortogonais. Como referido anteriormente, este sensor tem uma frequência de atualização de 100 Hz. O GPS utilizado tem a referência GPSlim236 (Figura A.5) e tem uma taxa de atualização de 1 Hz. As especificações completas de cada sensor, bem como os processos de calibração dos mesmos, encontram-se disponíveis em Apêndice, na Secção A.

Ambos os sensores, IMU e GPS, devolvem a orientação em função da direção do Norte, assumindo que não existe declinação magnética (ângulo formado entre o norte magnético e o norte verdadeiro) em que, ao contrário da convenção matemática, o ângulo é crescente no sentido horário e tem um desfasamento de  $90^\circ$  (ver Figura 4.8(a) e Figura 4.8(b)). A conversão do ângulo em função da orientação do Norte para o da convenção matemática,

consiste na inversão do ângulo e no deslocamento de  $-90^\circ$ , ou seja  $\theta_{mat} = -(\theta_{sensor} - 90^\circ)$ .



(a) Ângulo em função da orientação do Norte [35] .



(b) Ângulo na convenção matemática.

Figura 4.8: Ângulo obtido pelos sensores VS ângulo na convenção matemática

A comunicação entre os sensores e o sistema é feita através de um Arduino<sup>2</sup>, mais especificamente o Arduino mega 2560 (para mais especificações, e o respetivo esquema de ligação, consultar a Secção A.2.1). Este apenas tem a função de adquirir os sinais dos sensores e transmiti-los para a aplicação SLVA. A Figura 4.9 representa os tópicos de comunicação utilizados para os sensores em ambiente real. Como referido, mesmo estando implementados no veículo real, os atuadores não foram utilizados, com o intuito de avaliar o impacto no sistema, e também para ter um sistema preparado para a condução manual e autónoma. Nesta figura, o `roserial` e `can_serie` são aplicações desenvolvidas para fazer a interligação do *hardware* com o ROS, as quais são descritas mais à frente, no Apêndice B.1.2.

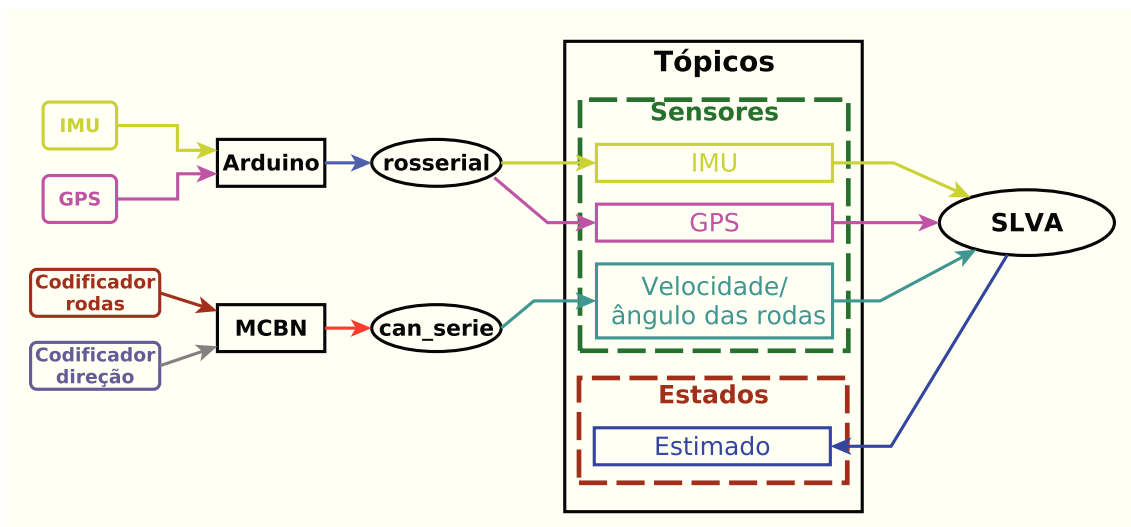


Figura 4.9: Tópicos de comunicação utilizados em ambiente real.

<sup>2</sup>[www.arduino.cc](http://www.arduino.cc)

## 4.4 Implementação dos filtros

Como a informação sensorial não é toda adquirida no mesmo instante nem com a mesma frequência, a implementação dos filtros teve que ter esse fator em conta, de modo a que seja processada individualmente aquando da sua disponibilização. Com frequências diferentes das utilizadas neste trabalho, para melhor interpretação, a Figura 4.10 representa os intervalos de tempo entre aquisições sensoriais ao longo do tempo e os processos executados para cada aquisição.

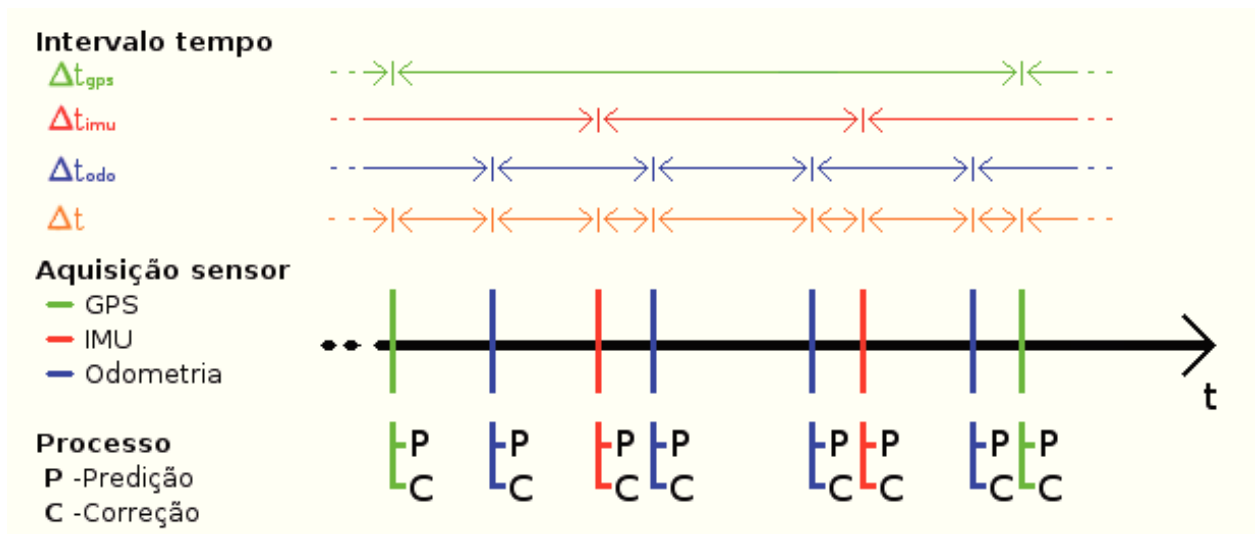


Figura 4.10: *Intervalos de tempo entre aquisições sensoriais.*

Na Figura 4.10,  $\Delta t$  corresponde ao intervalo de tempo entre predições, ou seja, tempo decorrido entre aquisições sensoriais;  $\Delta t_{gps}$ ,  $\Delta t_{imu}$  e  $\Delta t_{odo}$  representam o intervalo de tempo entre aquisições do GPS, da odometria e da IMU, respetivamente.

A Figura 4.10 refere-se a um exemplo para o EKF\_1, podendo extrapolar-se o raciocínio para os restantes estimadores implementados. Na mesma figura é possível observar que, antes de cada correção, é realizado o processo de predição. Isto deve-se ao facto de, no processo de correção/atualização, o vetor de medidas e o vetor de medidas estimadas terem de ser referentes ao mesmo instante de tempo.

### 4.4.1 EKF com a odometria no passo de correção

Para evitar dependências entre sensores, o vetor de medidas foi dividido por sensores. Neste filtro são utilizados os três sensores na atualização, pelo que o vetor de medidas será dividido em três vetores distintos, ou seja, um por sensor. Então, no caso de indisponibilidade da informação de algum sensor, é possível realizar na mesma o processo de atualização, apenas considerando os sensores disponibilizados.

- **Odometria:** O vetor de medidas para a odometria e o vetor de medidas estimadas são definidos pela Equação (4.3) e pela Equação (4.4), respectivamente, com a matriz do jacobiano do sistema dada pela Equação (4.5).

$$z_{k+1} = \begin{bmatrix} v_{odo} \\ \frac{v_{odo} \tan(\alpha)}{d_{eixo}} \\ \frac{v_{odo} - v_{last\_odo}}{\Delta t_{odo}} \end{bmatrix} \quad (4.3)$$

$$\hat{z}_{k+1} = h(\hat{X}_{k+1}^-) = \begin{bmatrix} \hat{v}_{k+1} \\ \hat{\omega}_{k+1} \\ \hat{a}_{k+1} \end{bmatrix} \quad (4.4)$$

$$H_{k+1} = \left. \frac{\partial h}{\partial X} \right|_{X=\hat{X}_{k+1}^-} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.5)$$

Onde  $v_{last\_odo}$  corresponde à velocidade linear adquirida pela odometria no instante anterior.

- **IMU:** O vetor de medidas referente à IMU e o vetor de medidas estimadas são definidos pela Equação (4.6) e pela Equação (4.7), respectivamente, com a matriz do jacobiano do sistema dada pela Equação (4.8).

$$z_{k+1} = \begin{bmatrix} \theta_k + \Delta\theta_{bussola} \\ \omega_{giro} \\ a_{acel} \end{bmatrix} \quad (4.6)$$

$$\hat{z}_{k+1} = h(\hat{X}_{k+1}^-) = \begin{bmatrix} \hat{\theta}_{k+1} \\ \hat{\omega}_{k+1} \\ \hat{a}_{k+1} \end{bmatrix} \quad (4.7)$$

$$H_{k+1} = \left. \frac{\partial h}{\partial X} \right|_{X=\hat{X}_{k+1}^-} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.8)$$

- **GPS:** O vetor de medidas para o GPS e o vetor de medidas estimadas são definidos pela Equação (4.9) e pela Equação (4.10), respectivamente, com a matriz do jacobiano do sistema dada pela Equação (4.11).

$$z_{k+1} = \begin{bmatrix} x_{gps} \\ y_{gps} \\ \theta_{gps} \\ v_{gps} \\ \frac{\theta_{gps} - last\_ \theta_{gps}}{\Delta t_{gps}} \\ \frac{v_{gps} - last\_ v_{gps}}{\Delta t_{gps}} \end{bmatrix} \quad (4.9)$$

$$\hat{z}_{k+1} = h(\hat{X}_{k+1}^-) = \begin{bmatrix} \hat{x}_{k+1} \\ \hat{y}_{k+1} \\ \hat{\theta}_{k+1} \\ \hat{v}_{k+1} \\ \hat{\omega}_{k+1} \\ \hat{a}_{k+1} \end{bmatrix} \quad (4.10)$$

$$H_{k+1} = \left. \frac{\partial h}{\partial X} \right|_{X=\hat{X}_{k+1}^-} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.11)$$

#### 4.4.2 EKF com a odometria no passo de predição

Como referido anteriormente, a fim de garantir que no processo de correção o vetor de medidas e o vetor de medidas estimadas sejam referentes ao mesmo instante de tempo, é necessário executar o processo de predição imediatamente antes do processo de correção. Neste método, o processo de predição está dependente da odometria. Assim, caso o instante temporal associado ao último valor de odometria não seja o mesmo do último valor do(s) sensor(es), neste caso estima-se o estado considerando um modelo puramente preditivo, considerando uma dinâmica constante, segundo a Equação (3.22).

Relativamente ao passo de correção, e à semelhança do caso anterior, o vetor de medidas foi também dividido por sensor, de forma a permitir a receção e correção em função da informação sensorial recebida.

#### 4.4.3 Filtro de partículas

O modelo baseado no PF sofre exatamente as mesmas alterações que o EKF\_2. No caso de o instante temporal associado ao último valor de odometria não ser o mesmo do último

valor do(s) sensor(es), o estado é estimado com base num modelo puramente preditivo, considerando uma dinâmica constante. Em relação ao processo de atualização, o vetor de medidas também é dividido por sensor, de modo a poder executar-se este processo em função da informação sensorial recebida.

Para obter melhores resultados de localização, no cálculo do peso das partículas em função da informação sensorial do GPS, a importância atribuída a cada parâmetro do estado é dada em função do erro da posição, ou seja, quanto mais afastada a partícula estimada estiver da localização medida pelo GPS, menor vai ser a importância dada aos restantes parâmetros do estado. Assim, as partículas que tiverem igual erro em posição, também terão a mesma importância associada a cada parâmetro, no cálculo do peso. Na Figura 4.11 é representada um exemplo para cinco partículas.

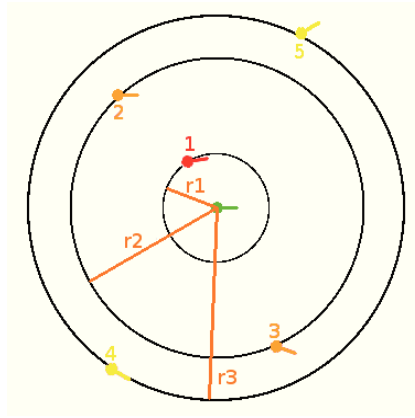


Figura 4.11: *Exemplo da influência da distância no cálculo do peso das partículas.*

Onde  $r_1$ ,  $r_2$  e  $r_3$  representam os erros da distância das partículas estimadas e a medida obtida pelo GPS; a medida estimada pelo GPS é representada a verde; e as partículas estimadas são representadas a cor amarela, laranja e vermelha, dependendo do fator de escala. A cor amarela representa uma importância mais baixa, a laranja uma intermédia e a vermelha a mais alta.

Como é possível observar na Figura 4.11, as partículas que estão à mesma distância do estado medido têm a mesma importância, nomeadamente comparando a partícula 2 com a 3 e a partícula 4 com a 5. Esta importância é calculada através de um fator de escala calculado em função do erro da distância da partícula à posição medida. O cálculo deste fator de escala é apresentado na Equação (4.12).

$$F\_escala = \frac{r^{(i)}}{\max(r)} \quad (4.12)$$

Onde  $F\_escala$  corresponde ao fator de escala;  $r^{(i)}$  à distância desde a posição medida à posição estimada, da respetiva partícula;  $\max(r)$  corresponde ao  $r^{(i)}$  de maior dimensão do conjunto de partículas. Por exemplo, para a partícula  $i = 3$  da Figura 4.11, o  $r^{(i)}$

corresponde à distância  $r_2$  e o  $\max(r)$  corresponde ao  $r_3$ .

Por forma a acelerar a convergência da localização estimada com a localização medida pelo GPS, quando o sistema está perdido no espaço, é feita uma reinicialização de 30% das partículas em torno da localização medida pelo GPS. Considerando que o sistema está perdido no espaço quando todas as partículas estão afastadas, pelo menos, cinco metros da localização medida pelo GPS, valor superior à variância do sensor (ver A.2.2), as partículas com menor peso são espalhadas aleatoriamente em torno da localização medida pelo GPS, num raio de cinco metros. São apenas espalhadas uma minoria das partículas devido a não ser utilizada nenhuma medida de qualidade do sinal do GPS. Ao não fazer total confiança neste sensor, previne-se os erros temporários/variações do GPS, corrigindo o sistema, quando este está perdido no espaço.

A implementação do processo de atualização através da IMU, na sua forma genérica, faz com que o cálculo do peso atribuído às partículas seja em função do seu deslocamento, não tendo assim qualquer tipo de referência absoluta. Assim, as partículas que tenham um menor erro absoluto, mas um maior erro relativo (deslocamento desde a última aquisição), terão um peso superior, e vice-versa. Esta situação origina que, no processo de reamostragem, as partículas que tenham um menor erro absoluto, mas um maior erro relativo (deslocamento desde a última aquisição), sejam descartadas e, na situação contrária, sejam duplicadas. O método utilizado para corrigir este problema, consistiu na definição de um sensor virtual de localização absoluta. Este sensor virtual corresponde à localização do GPS. Entre cada aquisição do GPS, este valor é atualizado com base na última localização do GPS e na odometria, considerando que esta é perfeita. O vetor de medidas e o vetor de medidas estimadas são definidos pelo sensor virtual e pela IMU - Equação (4.13) e Equação (4.14), respetivamente.

$$z_{k+1} = \begin{bmatrix} x_{virtual} \\ y_{virtual} \\ \theta_{virtual} \\ v_{virtual} \\ \omega_{virtual} \\ a_{virtual} \\ \theta_k + \Delta\theta_{bussola} \\ \omega_{giro} \\ a_{acel} \end{bmatrix} \quad (4.13)$$

$$\hat{z}_{k+1} = h(\hat{X}_{k+1}^-) = \begin{bmatrix} \hat{x}_{k+1} \\ \hat{y}_{k+1} \\ \hat{\theta}_{k+1} \\ \hat{v}_{k+1} \\ \hat{\omega}_{k+1} \\ \hat{a}_{k+1} \\ \hat{\theta}_{k+1} \\ \hat{\omega}_{k+1} \\ \hat{a}_{k+1} \end{bmatrix} \quad (4.14)$$

Onde  $x_{virtual}$  e  $y_{virtual}$  correspondem à posição em  $x$  e em  $y$ , respetivamente;  $\theta_{virtual}$  à orientação do veículo no instante atual ( $k + 1$ ), utilizando convenção matemática;  $v_{virtual}$  e  $\omega_{virtual}$  à velocidade linear e angular para o instante atual ( $k + 1$ ), respetivamente;  $a_{virtual}$  corresponde à aceleração linear no referencial do veículo, todos relativos ao sensor virtual.

# Capítulo 5

## Resultados

Neste capítulo são apresentados os resultados obtidos nos testes realizados em ambiente de simulação, seguindo-se os resultados dos testes realizados em ambiente real. Por último, é feita uma descrição e análise dos resultados obtidos.

### 5.1 Testes realizados

Para melhor comparar os algoritmos, considerando condições iguais, o veículo foi conduzido em modo manual no ambiente de simulação e no ambiente real, tendo sido gravados todos os valores lidos dos sensores. Assim, o percurso realizado, assim como os ruídos aleatórios adicionados no ambiente de simulação, utilizados para testar os vários algoritmos, são exatamente os mesmo.

Os testes em ambiente real foram realizados nos parques de estacionamento da Escola Superior de Tecnologia e Gestão (ESTG) de Leiria, estando por isso limitados à sua dimensão e à ocupação de espaço por veículos estacionados. Os testes em ambiente de simulação foram efetuados restringindo o espaço disponível para dimensões semelhantes às disponíveis no ambiente real. Para a avaliação de desempenho dos métodos de estimação do estado, foi definida uma referência. No ambiente de simulação, a referência considerada foi a localização real, disponibilizada pelo simulador (Gazebo). Já no ambiente real, a localização real não está disponível, pelo que foi considerado o GPS como referência de posição e orientação, e a odometria como referência da velocidade linear, visto serem os sensores com menor erro em média para cada uma das medidas, respetivamente.

Sabendo que a IMU é um sensor particularmente sensível a perturbações externas, tais como inclinações ou trepidações no caso dos acelerómetros e giroscópios e materiais metálicos, campos magnéticos produzidos por, por exemplo, condutores elétricos e dispositivos eletrónicos, no caso dos magnetómetros, são realizados os mesmos testes utilizando

e não utilizando a IMU, ficando assim a saber qual a sua influência sobre o sistema.

Ao retirar a IMU no modelo PF, a frequência de atualização é a mesma do GPS, que é muito baixa (ver Figura 5.1(a)). Para efeitos de controlo, foi adicionada ao PF a funcionalidade de estimar o estado no processo de predição, ficando assim com uma maior frequência de estimação, ver Figura 5.1(b). Esta estimativa consiste na predição do estado em função do estado anterior e do sinal de controlo, considerando que a odometria é perfeita. De referir que esta funcionalidade não altera em nada o sistema, apenas estima o estado para possíveis efeitos de controlo. Pode assim ser observado que os espaços vazios entre aquisição do GPS, representantes do desconhecimento do estado, são preenchidos na segunda situação.

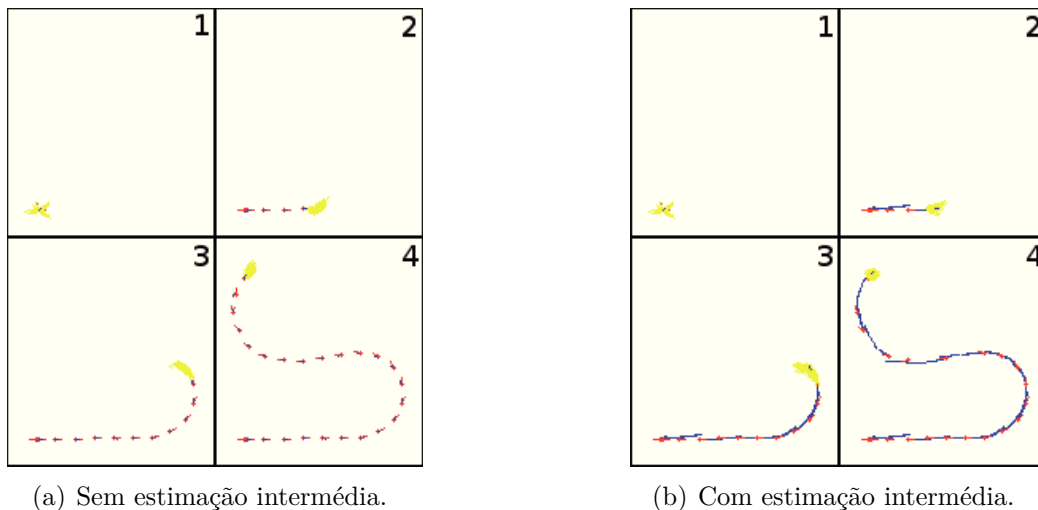


Figura 5.1: Comparação do modelo PF com e sem estimação do estado na predição

A trajetória estimada pelo método em estudo é representada a azul; a trajetória adquirida pelo GPS a vermelho; a cor amarela representa as várias partículas estimadas. Afim de melhorar a interpretação do modelo PF, foi reduzida a informação de partículas estimadas apresentadas durante a evolução do percurso, ver Figura 5.1. Como é possível observar, as partículas estimadas referentes aos estados anteriores são descartadas, apresentando apenas as partículas estimadas referentes ao instante atual, representadas em amarelo.

Os testes efetuados para ambos os ambientes, consistiram num percurso típico de uma elipse e num percurso com a forma de um oito. No caso do ambiente real, foi também efetuado um teste que consistiu no percurso entre dois parques de estacionamento da ESTG. Na Tabela 5.1 é apresentada a lista dos testes efetuados.

Tabela 5.1: *Lista de testes efetuados.*

	Teste	IMU	Método	Página
Ambiente de Simulação	Elipse	Com	EKF_1	51
			EKF_2	
			PF	
		Sem	EKF_1	55
			EKF_2	
			PF	
	Oito	Com	EKF_1	61
			EKF_2	
			PF	
Sem		EKF_1	65	
		EKF_2		
		PF		
Ambiente Real	Elipse	Com	EKF_1	70
			EKF_2	
			PF	
		Sem	EKF_1	74
			EKF_2	
			PF	
	Oito	Com	EKF_1	79
			EKF_2	
			PF	
		Sem	EKF_1	82
			EKF_2	
			PF	
Normal	Com	EKF_1	87	
		EKF_2		
		PF		
	Sem	EKF_1	91	
		EKF_2		
		PF		

### 5.1.1 Ambiente de simulação

Os testes em ambiente de simulação foram efetuados impondo uma velocidade linear constante ao veículo e controlando manualmente o ângulo do volante. Como referido, o ambiente de simulação não tem restrições espaciais, tendo no entanto sido realizados

os testes assumindo restrições semelhantes ao ambiente real, nomeadamente no espaço disponível.

Para cada teste efetuado são adquiridas simultaneamente a localização, orientação e velocidade linear reais a partir do simulador, podendo assim calcular o erro absoluto dos vários métodos testados.

### Teste tipo elipse

O primeiro teste realizado consistiu num percurso em forma de elipse (aproximada). O teste foi realizado a uma velocidade constante de 3.5 m/s e teve como resultante uma elipse com uma largura de aproximadamente 115 m.

Sendo que o GPS e a odometria foram consideradas como referência no ambiente real, é importante apresentar o erro obtido, nos respetivos parâmetros, ao longo do percurso, por forma a ter uma noção da dimensão e relação com os resultados dos testes, visto os testes serem semelhantes aos executados no ambiente real. O erro obtido ao longo do tempo, está apresentado na Figura 5.2, na Figura 5.3 e na Figura 5.4, respetivamente em relação à posição e orientação do GPS e à velocidade linear da odometria. Na Tabela 5.2 está representado o desvio padrão e a média do erro obtido pelo GPS para este teste.

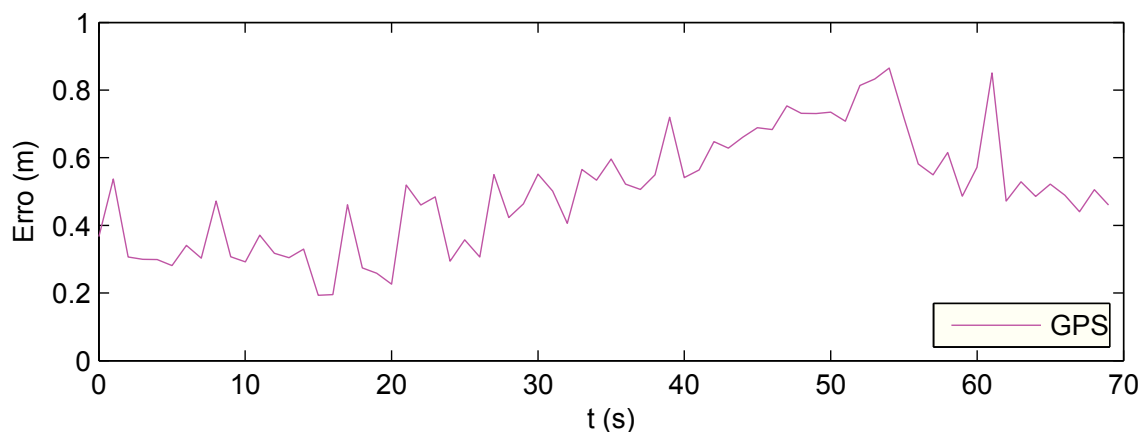


Figura 5.2: *Erro do GPS, em posição, ao longo do percurso tipo elipse.*

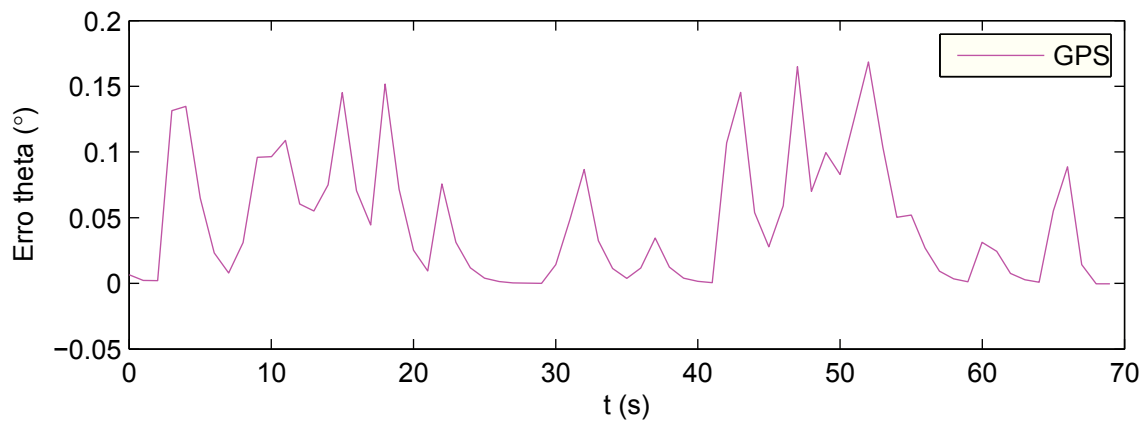


Figura 5.3: Erro do GPS, em orientação, ao longo do percurso tipo elipse.

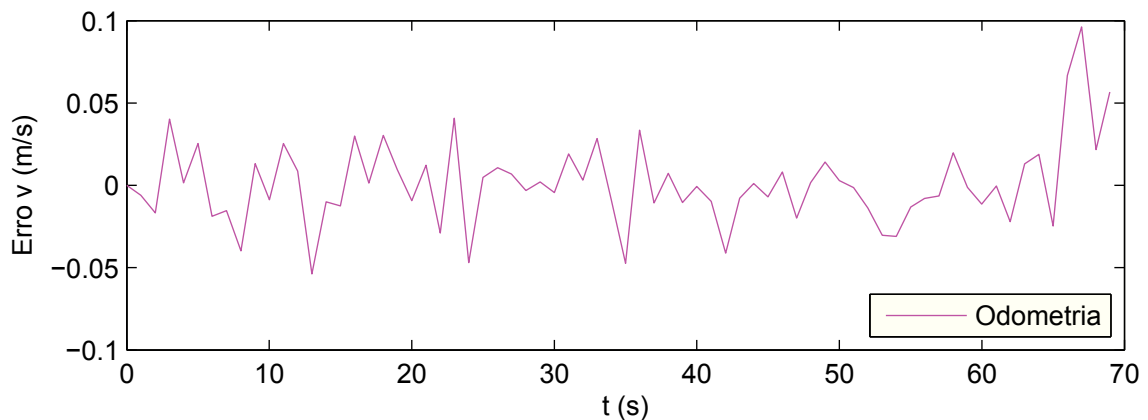


Figura 5.4: Erro da odometria, em velocidade, ao longo do percurso tipo elipse.

Tabela 5.2: O desvio padrão e média do erro dos sensores como referência, no teste tipo elipse.

	Desvio Padrão	Erro médio	Un
<b>Posição</b>	0.03531	0.18791	$m$
<b>Orientação</b> ( $\theta_{gps}$ )	2.73151	2.62923	$^{\circ}$ (grau)
<b>Velocidade linear</b> ( $v_{odo}$ )	0.02708	-0.00016	$m/s$

#### - Teste tipo elipse utilizando Odometria, GPS e IMU

Na Figura 5.5 é feita a comparação do percurso estimado, pelos diferentes métodos em análise neste trabalho, para o percurso do tipo elipse. A Figura 5.5(a) representa o teste realizado utilizando o método EKF\_1, a Figura 5.5(b) representa o método EKF\_2 e a Figura 5.5(c) o método PF.

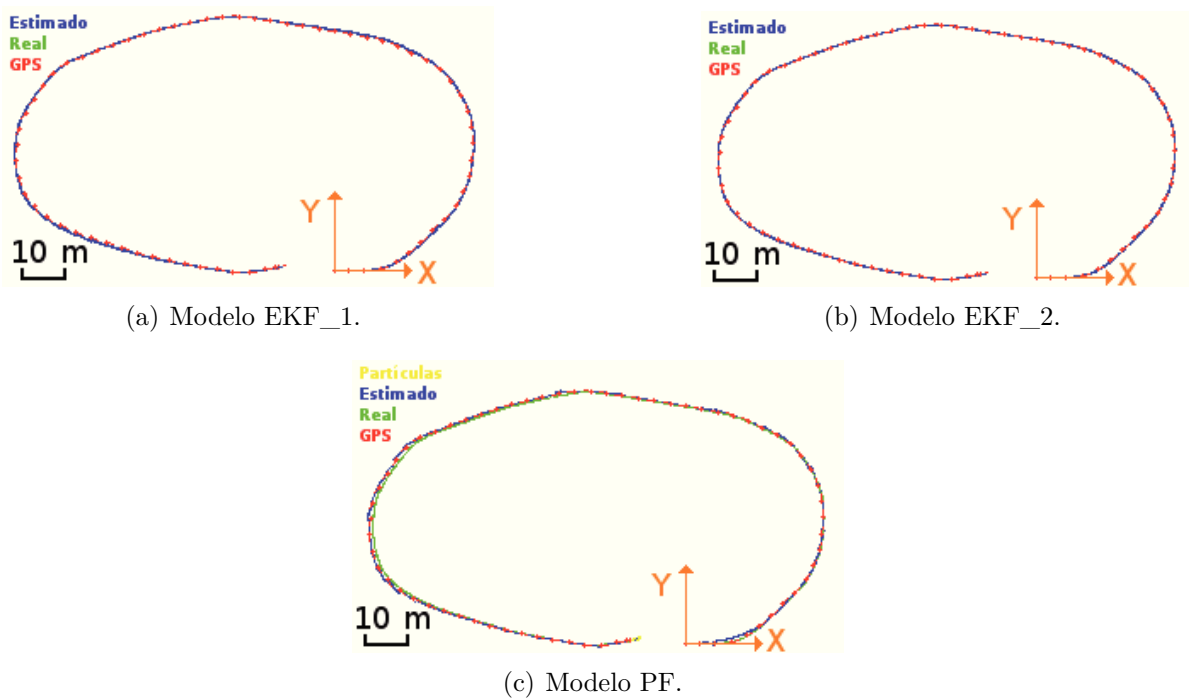


Figura 5.5: *Teste em simulação do tipo elipse, utilizando todos os sensores.*

A trajetória estimada pelo método em estudo é representada a azul; a trajetória adquirida pelo GPS a vermelho; a trajetória real a verde; e, no PF, a cor amarela representa as várias partículas estimadas. Note-se que a trajetória real não é visível devido a estar sobreposta pelas restantes trajetórias. É também visível nas Figuras o sistema de eixos global. Este é definido na posição inicial do veículo, orientado com o Y para Norte.

Para melhor comparação dos vários métodos, mostra-se de seguida o gráfico dos principais parâmetros do estado, nomeadamente  $x_{k+1}$ ,  $y_{k+1}$ ,  $\theta_{k+1}$  e  $v_{k+1}$  representados na Figura 5.6, Figura 5.7, Figura 5.8 e Figura 5.9, respetivamente.

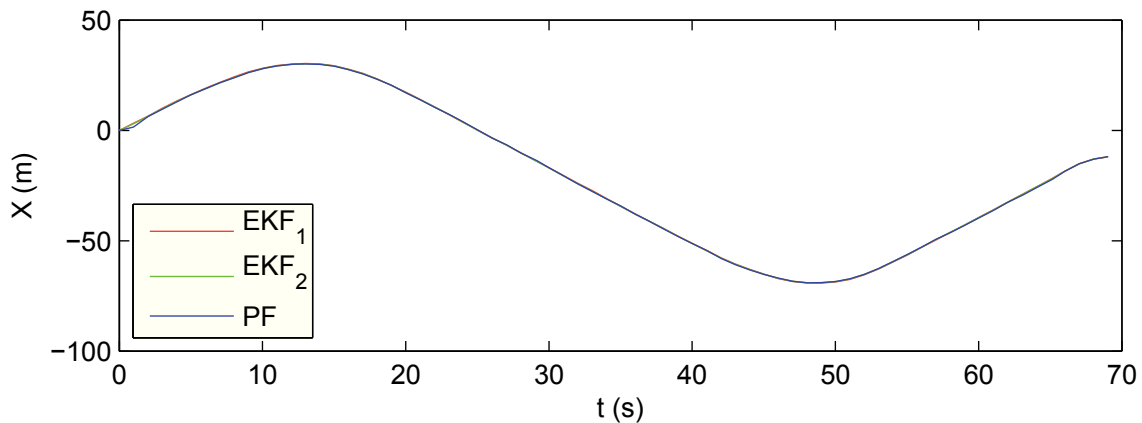


Figura 5.6: *Evolução da posição em X ao longo do percurso tipo elipse.*

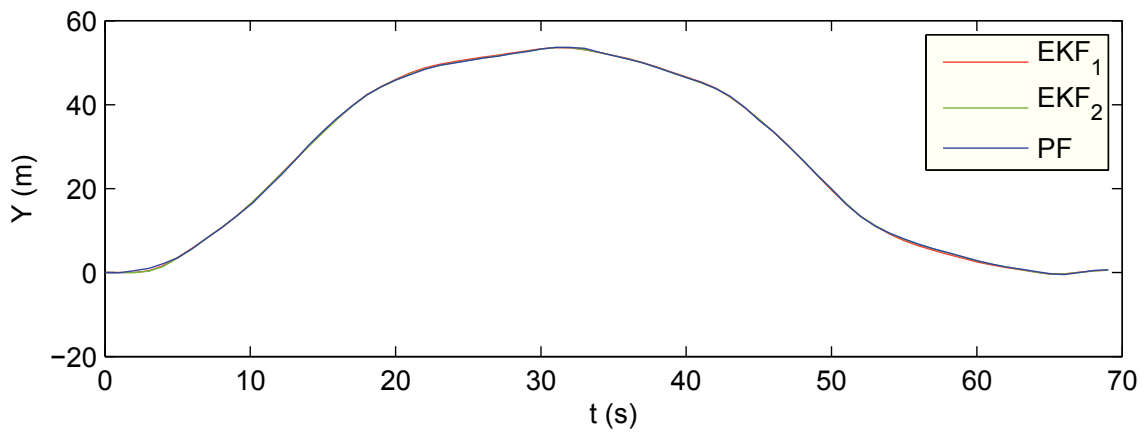


Figura 5.7: *Evolução da posição em Y ao longo do percurso tipo elipse.*

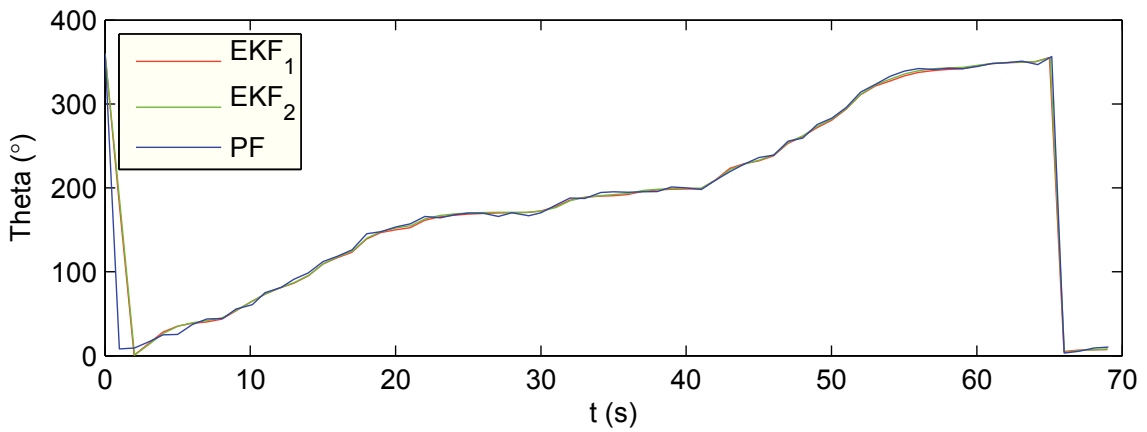


Figura 5.8: *Evolução da orientação ao longo do percurso tipo elipse.*

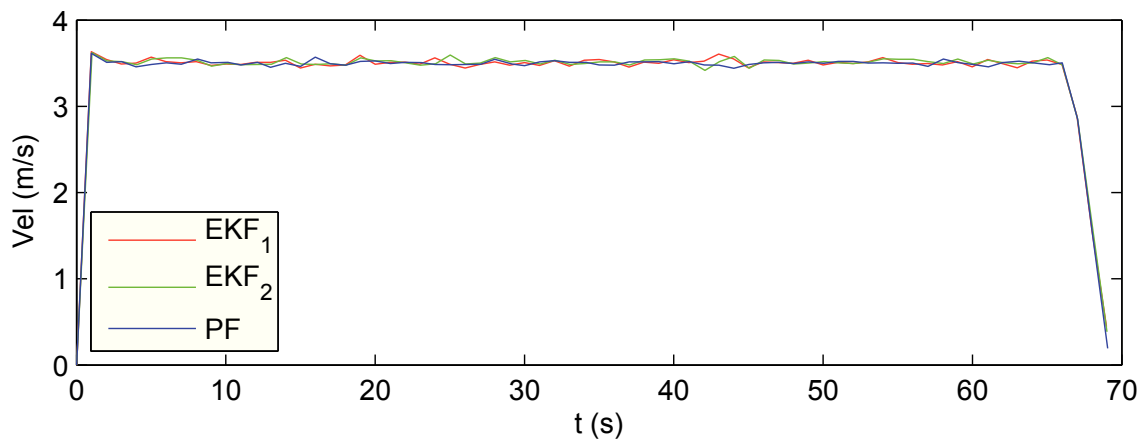


Figura 5.9: *Evolução da velocidade linear longo do percurso tipo elipse.*

Devido à semelhança dos resultados obtidos, em muitas situações não é possível visualizar alguns métodos devido a estarem sobrepostos. Para melhor comparação, é apresentado o erro obtido em termos de posição, orientação e velocidade linear na Figura 5.10, Figura 5.11 e Figura 5.12, respetivamente.

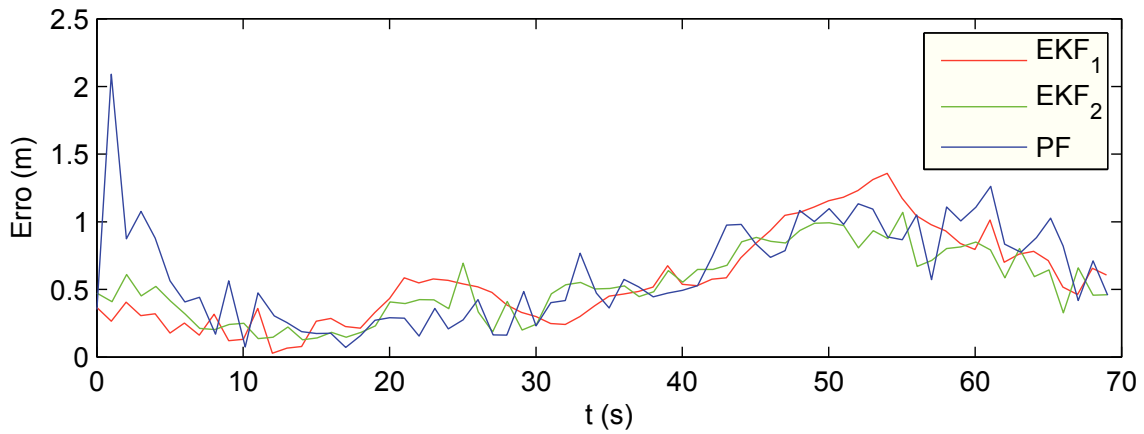


Figura 5.10: *Erro de posição ao longo do percurso tipo elipse.*

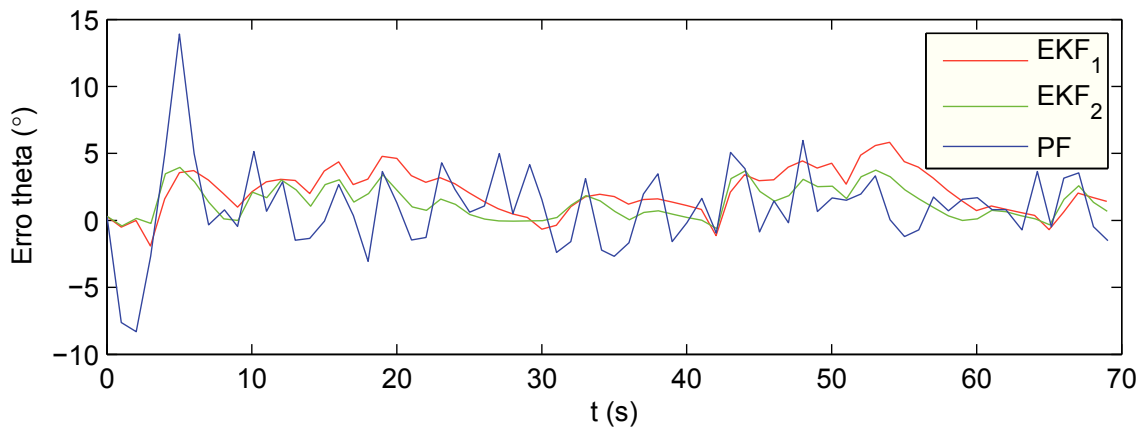


Figura 5.11: *Erro de orientação ao longo do percurso tipo elipse.*

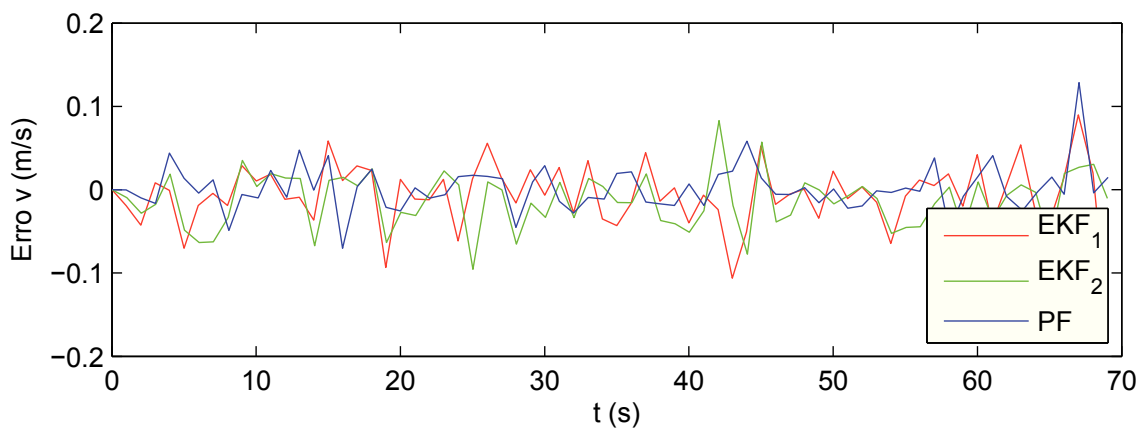


Figura 5.12: *Erro de velocidade ao longo do percurso tipo elipse.*

Como era de prever, o erro de posição obtido para o estado estimado vai de encontro ao erro obtido no GPS. Sendo que o GPS demonstrou ser o sensor com menor erro, dos três utilizados, foi-lhe atribuída maior confiança, fazendo assim a estimação do estado

convergir para o mesmo. Na Figura 5.13 é apresentado o erro em posição da estimativa do estado relativamente ao GPS. Isto é efetuado pois em ambiente real, não tendo medidas reais, é utilizado o GPS como referência, ficando assim com uma comparação mais fidedigna entre o ambiente real e o de simulação, com uma melhor noção da influência do erro do GPS no sistema. No caso do erro da orientação, assumindo um valor intermédio, é possível perceber que o erro é praticamente constante, variando ligeiramente quando são feitas alterações mais bruscas de orientação, comparando com a Figura 5.8. No erro da velocidade linear, o PF demonstrou ter um tempo de estabilização superior sendo, depois de estabilizado, semelhante ao EKF\_1 e ao EKF\_2. Na Tabela 5.3 é feita a comparação dos métodos em estudo em termos de posição, orientação e velocidade linear, para o percurso do tipo elipse.

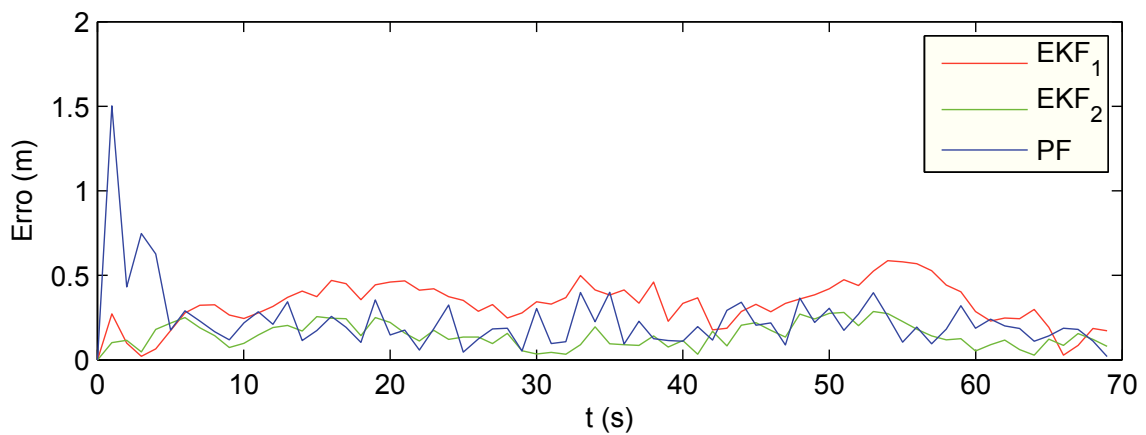
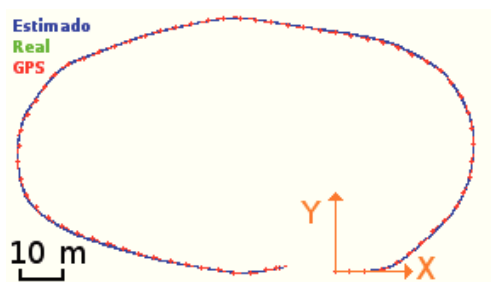


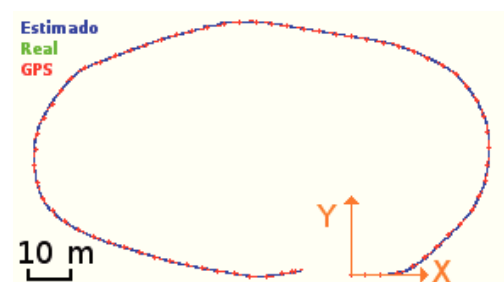
Figura 5.13: Erro de posição ao longo do percurso tipo elipse tendo o GPS como referência.

#### - Teste tipo elipse utilizando Odometria e GPS, não utilizando a IMU

A Figura 5.14 corresponde ao mesmo teste que o caso anterior, mas não utilizando a IMU. Nesta comparação, a Figura 5.14(a) representa o teste realizado utilizando o método EKF\_1, a Figura 5.14(b) utilizando o método EKF\_2 e a Figura 5.14(a) utilizando o método PF.

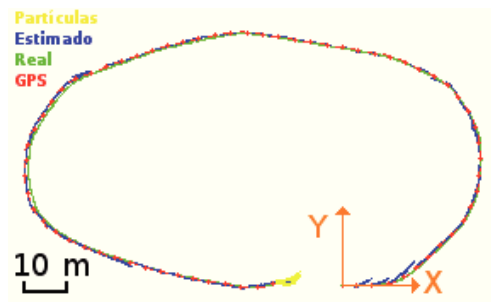


(a) Modelo EKF\_1.



(b) Modelo EKF\_2.

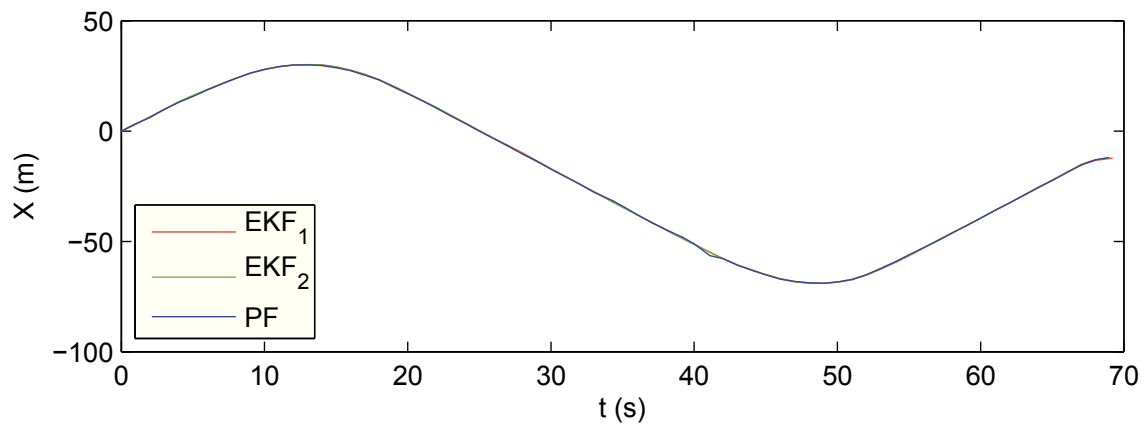
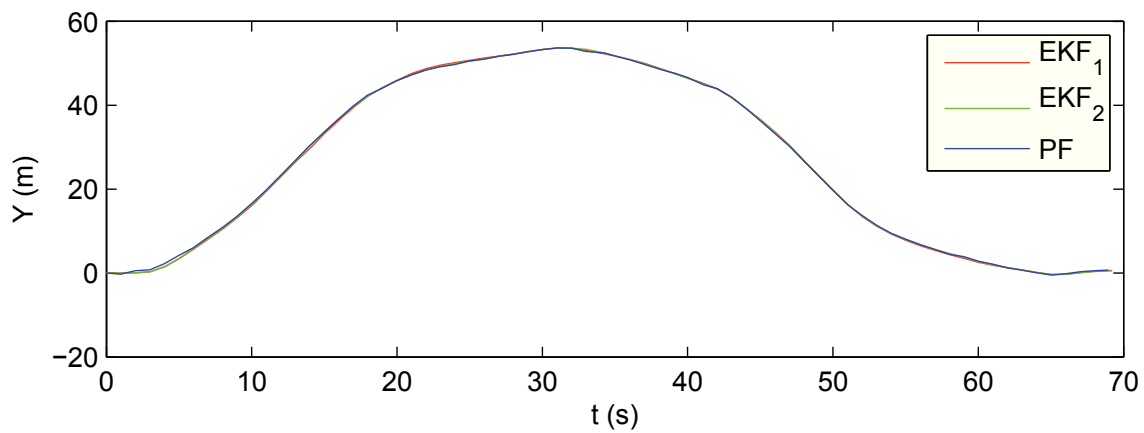
Figura 5.14: Teste em simulação do tipo elipse, não utilizando a IMU.



(a) Modelo PF.

Figura 5.14: *Teste em simulação do tipo elipse, não utilizando a IMU.*

De novo são apresentados os principais parâmetros do estado em gráfico, sendo a Figura 5.15, a Figura 5.16, a Figura 5.17 e a Figura 5.18 correspondentes a  $x_{k+1}$ ,  $y_{k+1}$ ,  $\theta_{k+1}$  e  $v_{k+1}$ , respectivamente.

Figura 5.15: *Evolução da posição em X ao longo do percurso tipo elipse.*Figura 5.16: *Evolução da posição em Y ao longo do percurso tipo elipse.*

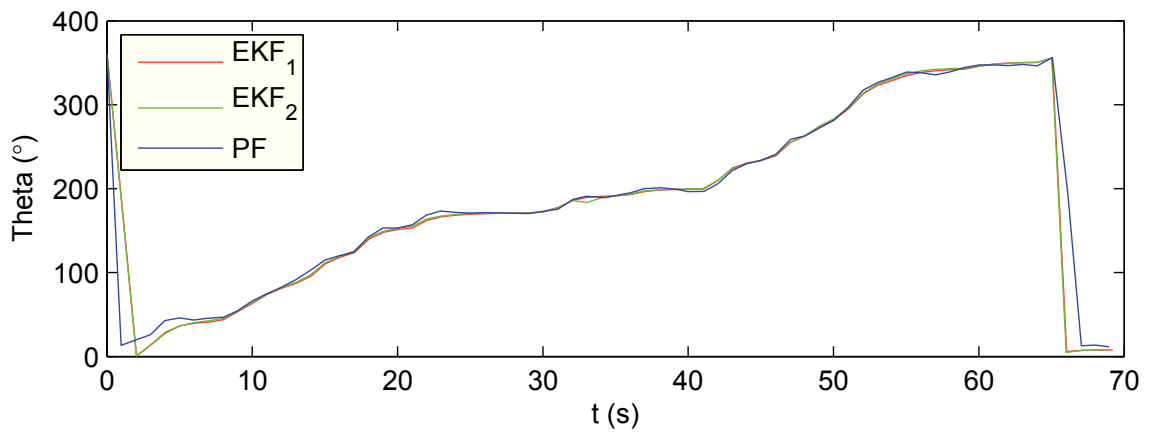


Figura 5.17: *Evolução da orientação ao longo do percurso tipo elipse.*

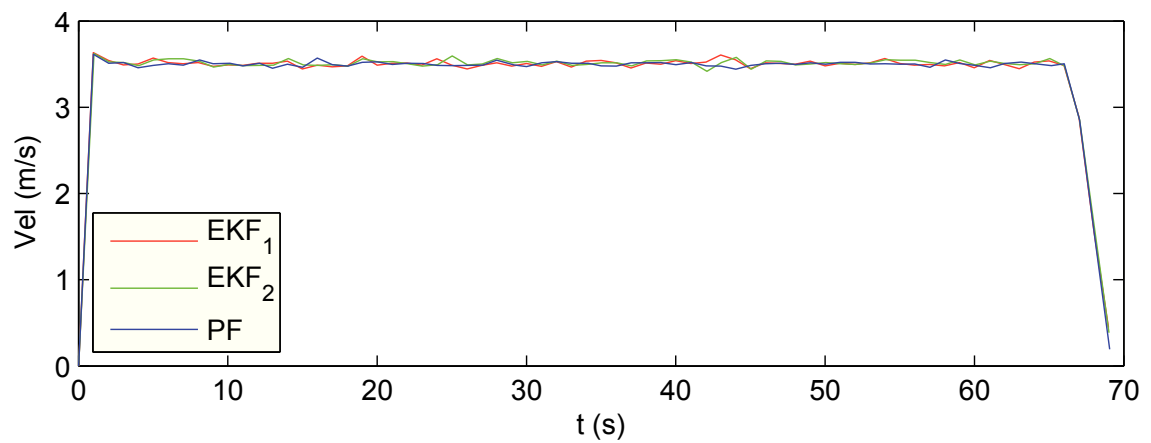


Figura 5.18: *Evolução da velocidade linear ao longo do percurso tipo elipse.*

Mais uma vez os resultados são bastante semelhantes, sendo inclusive difícil distingui-los devido à sobreposição. Para que seja possível fazer a comparação, é apresentado o erro obtido em termos de posição, orientação e velocidade linear na Figura 5.19, Figura 5.20 e Figura 5.21, respetivamente.

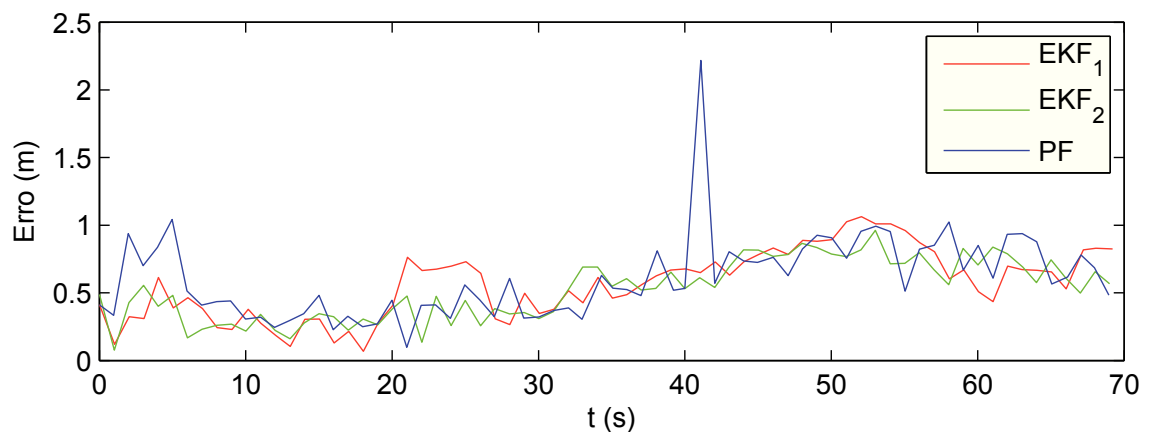


Figura 5.19: *Erro de posição ao longo do percurso tipo elipse.*

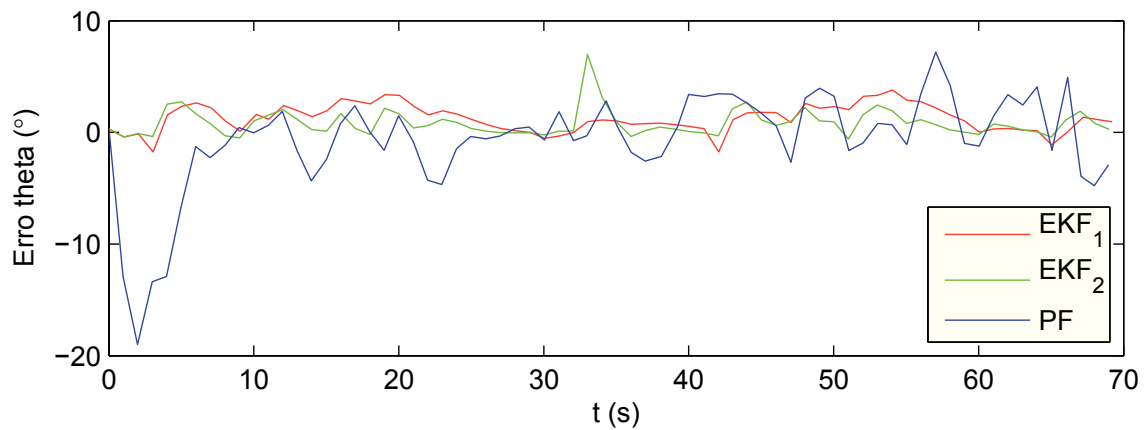


Figura 5.20: Erro da orientação ao longo do percurso tipo elipse.

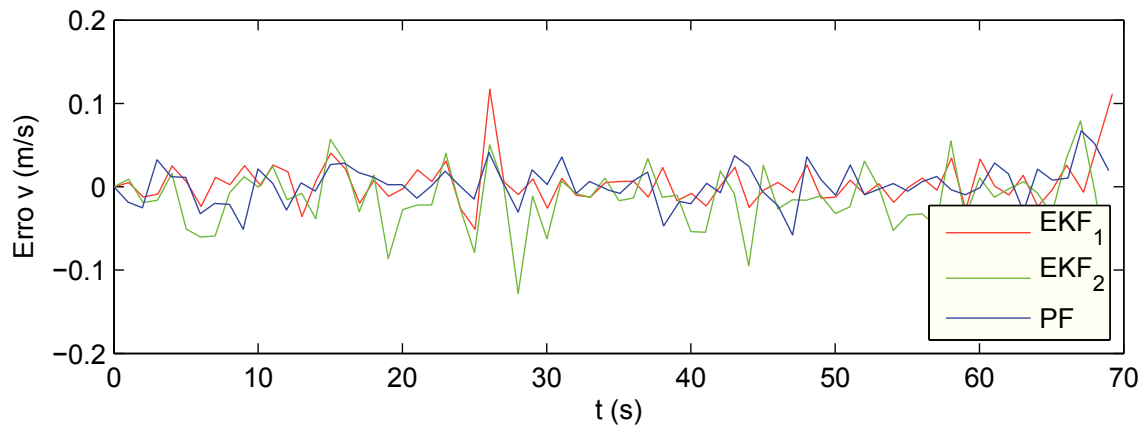


Figura 5.21: Erro da velocidade linear ao longo do percurso tipo elipse.

Os resultados obtidos foram relativamente semelhantes ao teste onde é utilizada a IMU. De novo, o erro de posição obtido para o estado estimado vai ao encontro do erro obtido no GPS, sendo por isso apresentado na Figura 5.22 o erro em posição, considerando o GPS como referência. Aqui, aos 40 segundos do método PF, é obtido um pico de distância de erro que leva a considerar que o sistema fica menos robusto sem a utilização da IMU, no teste em questão. Na orientação, o PF continua a demonstrar ter uma maior instabilidade, tanto no início do teste como durante o percurso. Ao contrário do teste anterior, o PF teve um tempo de estabilização de velocidade linear semelhante ao EKF\_1 e EKF\_2.

Na Tabela 5.3 é feita a comparação dos métodos em estudo em termos de posição, orientação e velocidade linear, para o percurso do tipo elipse em ambiente de simulação.

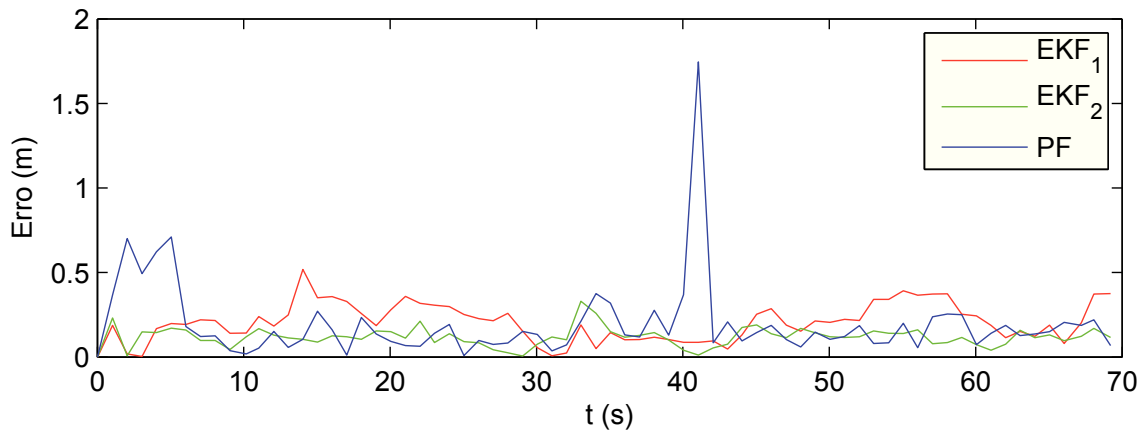


Figura 5.22: Erro de posição ao longo do percurso tipo elipse tendo o GPS como referência.

### Comparação para o percurso tipo elipse em simulação:

Na Tabela 5.3 está representado o desvio padrão e a média do erro, obtidos para os métodos em estudo, no percurso da elipse, fazendo a distinção entre a utilização e a não utilização da IMU. A vermelho está representado o método que apresentou melhores resultados de posição, a verde de orientação e a azul de velocidade linear.

Tabela 5.3: Comparação entre algoritmos na simulação para o percurso do tipo elipse.

	Método	Utilizando IMU		Não utilizando IMU		Un
		Desvio Padrão	Erro médio	Desvio Padrão	Erro médio	
Posição	EKF_1	0.33512	0.57325	0.24829	0.56789	$m$
	EKF_2	0.25809	0.53278	0.21800	0.52680	
	PF	0.41516	0.64939	0.38965	0.62285	
Orientação	EKF_1	1.65558	2.14652	3.17237	-0.98701	° (grau)
	EKF_2	1.28585	1.34738	3.01461	-1.17485	
	PF	3.94612	0.95794	4.46000	-2.00671	
Velocidade linear	EKF_1	0.69518	-0.06549	0.69525	-0.05448	$m/s$
	EKF_2	0.69441	-0.07207	0.69480	-0.07314	
	PF	0.79452	-0.08989	0.69466	-0.05629	

É possível observar que, de uma modo geral, em simulação, a IMU não trouxe benefícios para o sistema, tendo obtido em quase todos os resultados valores de erro em média superiores, aquando a sua utilização. Em relação à posição, ambos os testes tiveram resultados bastante semelhantes em erro médio, destacando-se entre eles o EKF\_2 sem utilização da IMU por ter o menor erro médio e um baixo Desvio Padrão. Na orientação, o erro médio também é próximo, comparando entre métodos, tendo sido o EKF\_2 com utilização da IMU o método que obteve melhor relação média - Desvio Padrão. Na velocidade também não existem diferenças significativas de erro médio entre os testes, tendo-se demonstrado maior precisão o EKF\_1 sem utilização da IMU. Pode-se concluir

que, no teste do tipo elipse em ambiente de simulação, o EKF\_2 foi sem dúvida o método que obteve melhores resultados. A utilização, ou não, da IMU vai de encontro com os critérios de escolha, neste caso em particular, os resultados foram tão próximos que seria imprudente fazer a escolha apenas baseada nos mesmos.

### Teste tipo oito

O segundo teste foi efetuado com um percurso em forma de oito. Este foi realizado a uma velocidade constante de 4.2 m/s, ficando o aspeto final com uma largura de aproximadamente 90 m.

De novo, é apresentado o erro obtido pelo GPS e pela odometria devido à sua utilização como referência no ambiente real. A Figura 5.23 e a Figura 5.24 representam o erro obtido pelo GPS em posição e orientação, enquanto que a Figura 5.25 representa o erro da odometria em velocidade linear. O desvio padrão e a média do erro, são apresentados na Tabela 5.4.

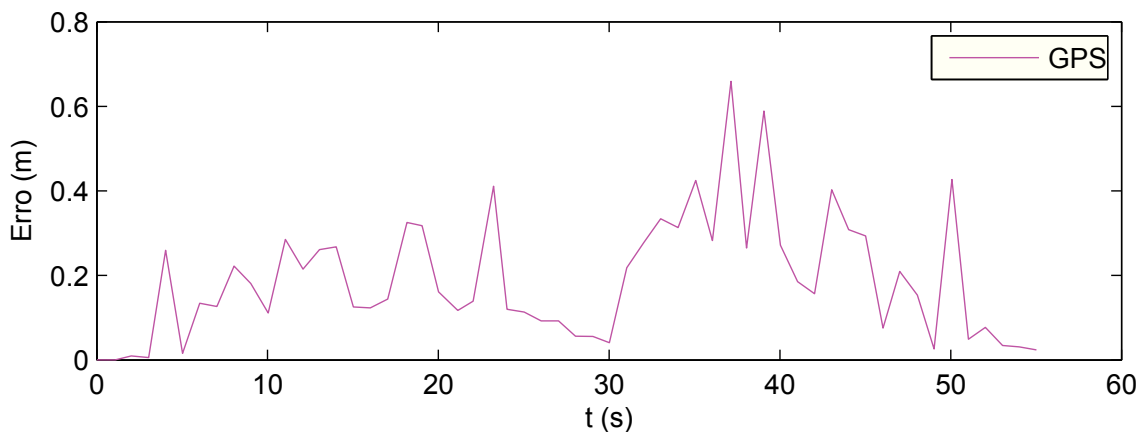


Figura 5.23: Erro do GPS em posição ao longo do percurso tipo oito.

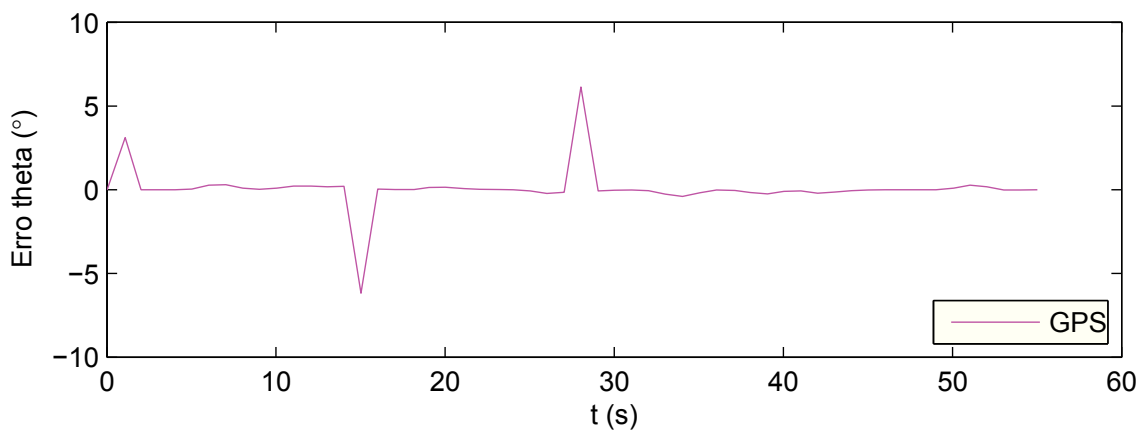


Figura 5.24: Erro do GPS em orientação ao longo do percurso tipo oito.

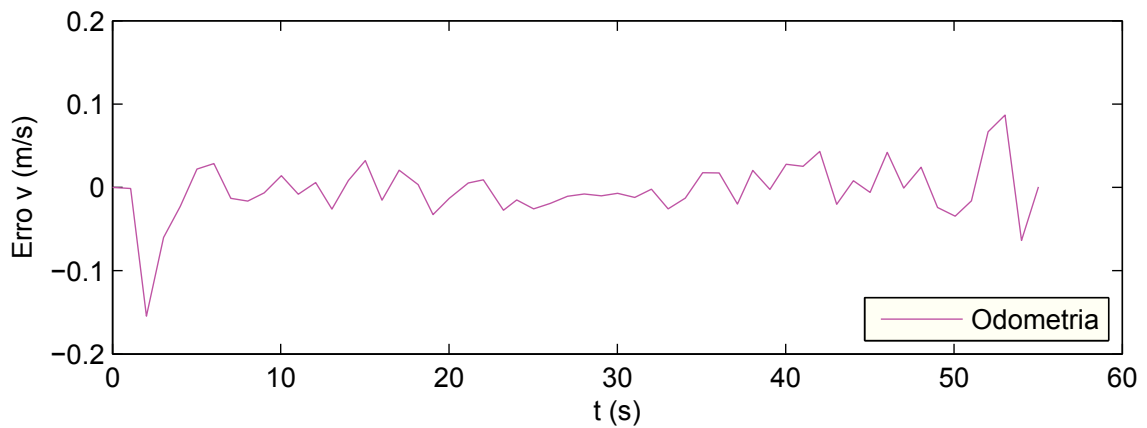


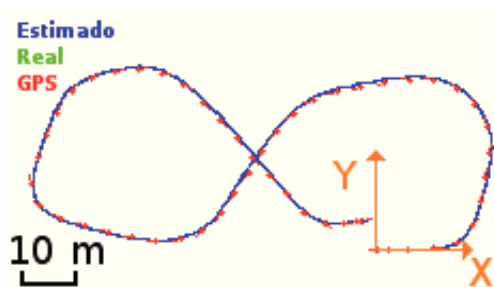
Figura 5.25: Erro da odometria em velocidade linear ao longo do percurso tipo oito.

Tabela 5.4: O desvio padrão e média do erro dos sensores como referência, no teste tipo oito.

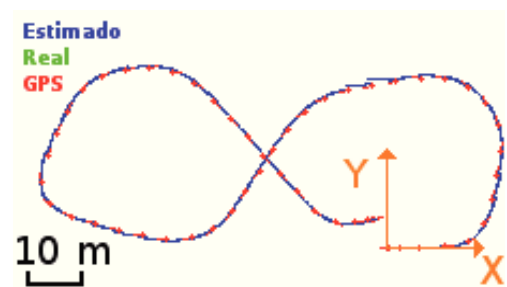
	Desvio Padrão	Erro médio	Un
<b>Posição</b>	0.15864	0.19648	$m$
<b>Orientação</b> ( $\theta_{gps}$ )	7.92752	0.10467	$^{\circ}$ (grau)
<b>Velocidade linear</b> ( $v_{odo}$ )	0.03425	-0.00593	$m/s$

#### - Teste tipo oito utilizando Odometria, GPS e IMU

A Figura 5.26(a) representa o teste realizado utilizando o EKF\_1, a Figura 5.26(b) com o EKF\_2 e a Figura 5.26(a) o PF.

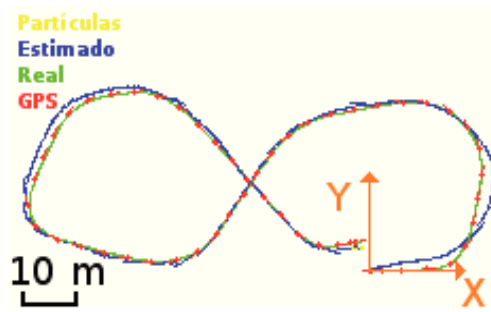


(a) Modelo EKF\_1.



(b) Modelo EKF\_2.

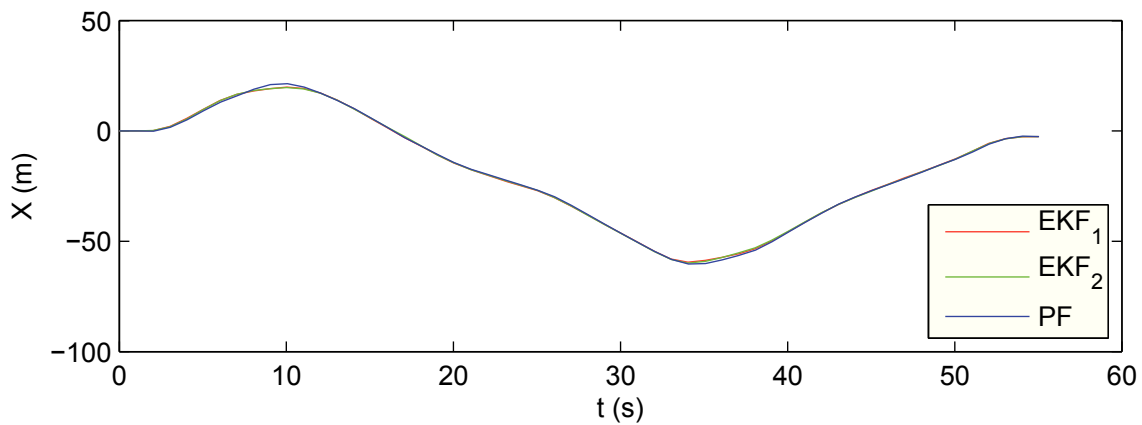
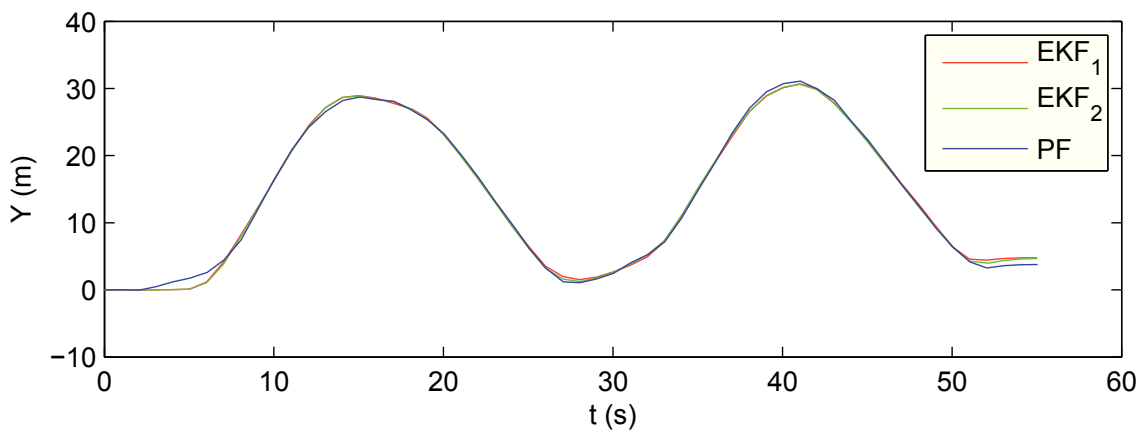
Figura 5.26: Teste em simulação do tipo oito.



(a) Modelo PF.

Figura 5.26: *Teste em simulação do tipo oito.*

De novo, para melhor comparação dos métodos são representados em gráfico os parâmetros do estado. Na Figura 5.27, Figura 5.28, a Figura 5.29 e a Figura 5.30, referente a  $x_{k+1}$ ,  $y_{k+1}$ ,  $\theta_{k+1}$  e  $v_{k+1}$ , respetivamente, de cada método.

Figura 5.27: *Evolução da posição em X ao longo do percurso tipo oito.*Figura 5.28: *Evolução da posição em Y ao longo do percurso tipo oito.*

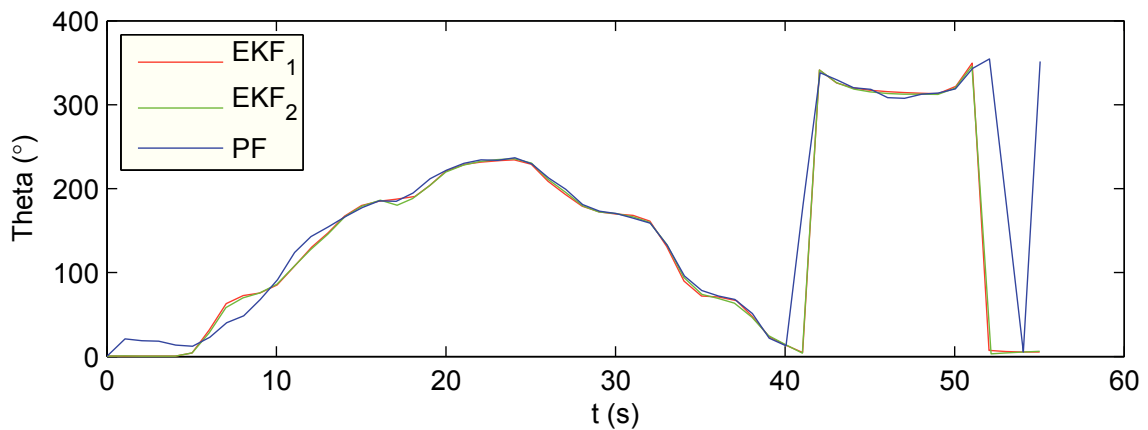


Figura 5.29: *Evolução da orientação ao longo do percurso tipo oito.*

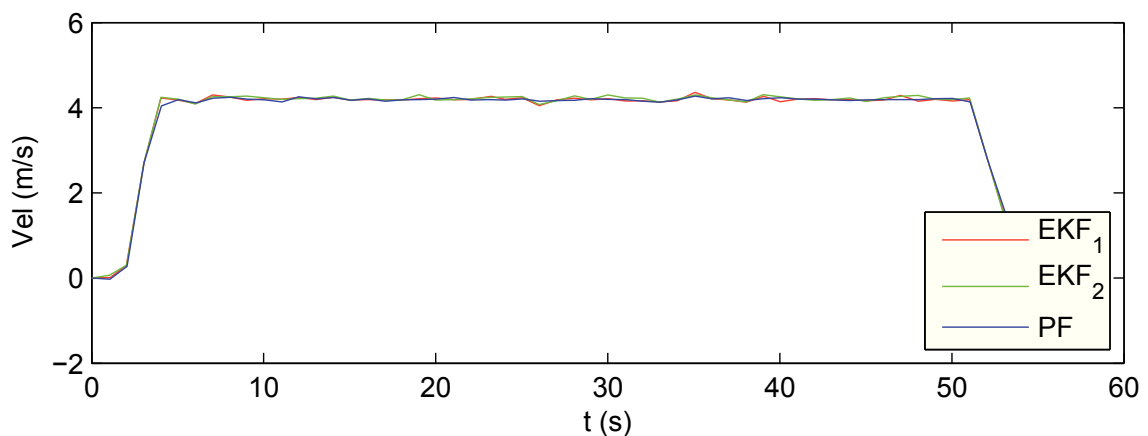


Figura 5.30: *Evolução da velocidade linear ao longo do percurso tipo oito.*

Mesmo existindo maiores diferenças entre os resultados obtidos pelos métodos, em comparação com os testes em elipse, em muitas situações não é possível distinguir alguns métodos devido à sua sobreposição. Para uma melhor diferenciação, são apresentados os erros obtidos em termos de posição, orientação e velocidade linear na Figura 5.31, na Figura 5.32 e na Figura 5.33, respectivamente em uma escala adaptada ao pretendido.

Neste teste, o PF destacou-se pela negativa, tendo obtido erros de posição e de orientação bastante superiores ao EKF\_1 e EKF\_2. Já o EKF\_1 e o EKF\_2 demonstraram ter valores bastante próximos em ambos os resultados. Mais uma vez, o erro em posição da estimativa é influenciado pelo erro em posição do GPS. Na Figura 5.34 estão representadas as evoluções do erro da posição ao longo do tempo para o mesmo percurso, quando utilizado o GPS como referência.

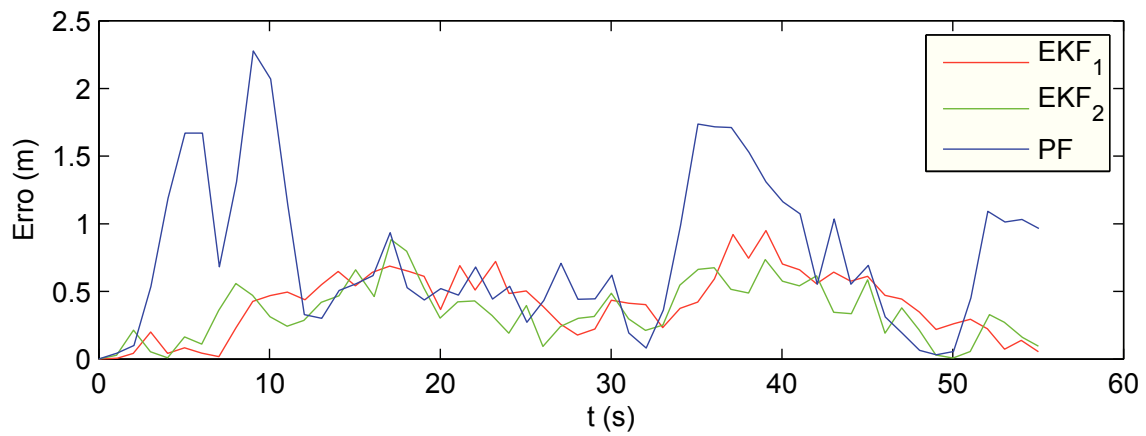


Figura 5.31: Erro de posição ao longo do percurso tipo oito.

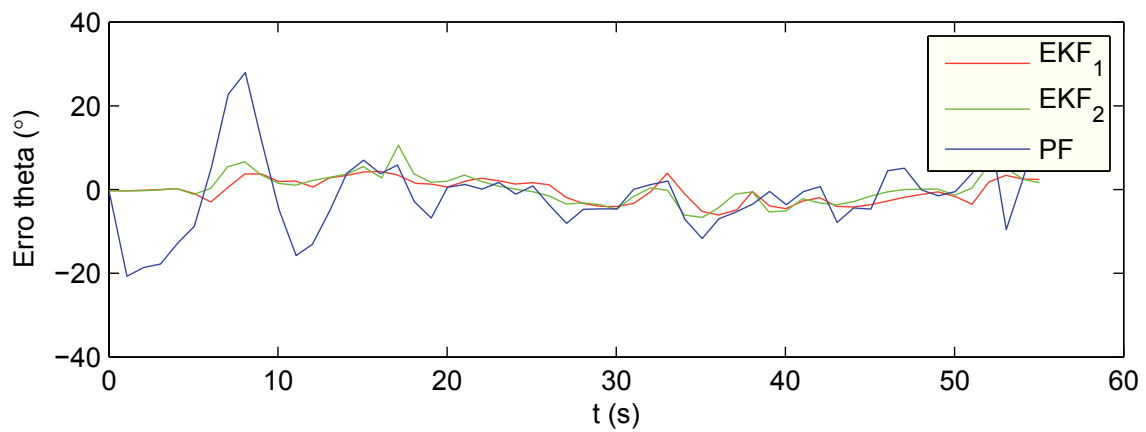


Figura 5.32: Erro de orientação ao longo do percurso tipo oito.

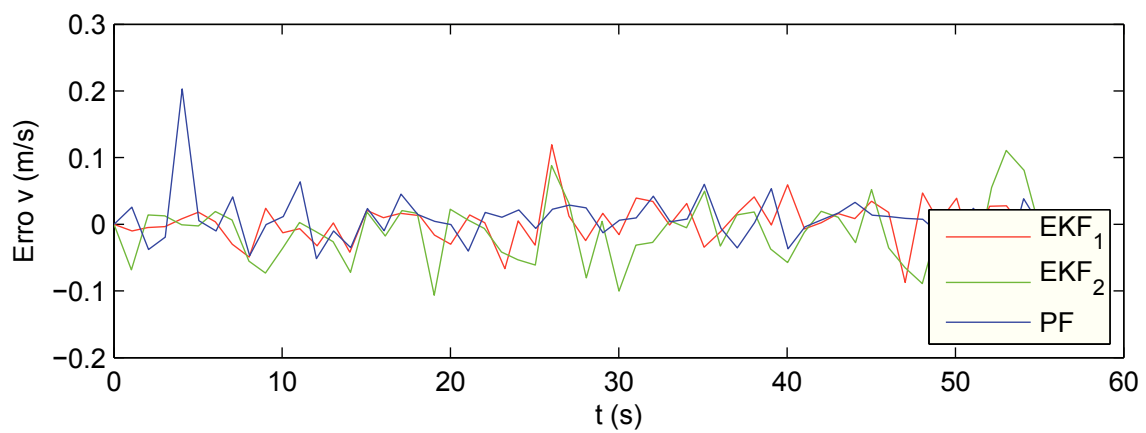


Figura 5.33: Erro de velocidade ao longo do percurso tipo oito.

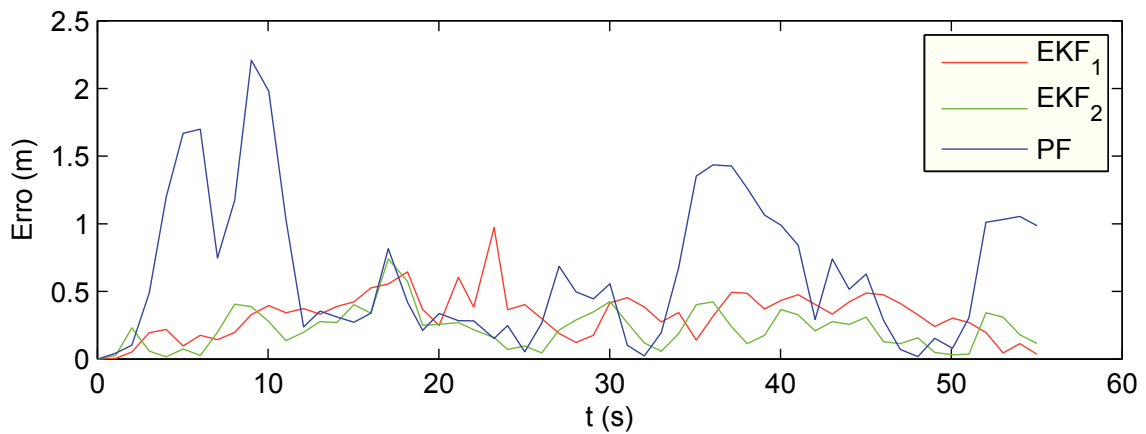
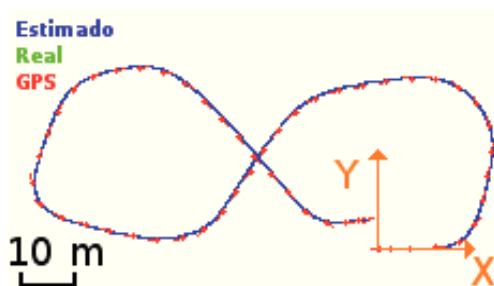


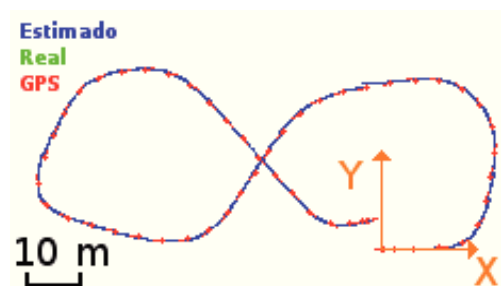
Figura 5.34: Erro de distância ao longo do percurso tipo oito tendo o GPS como referência.

- Teste tipo oito utilizando Odometria e GPS, não utilizando a IMU

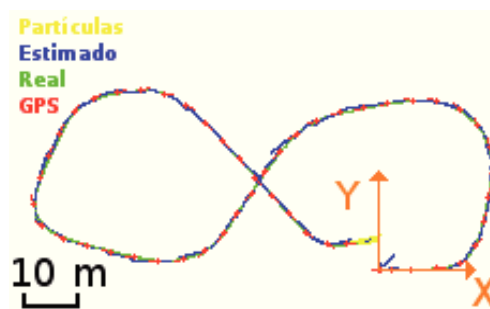
A Figura 5.35(a) representa o teste realizado utilizando o EKF\_1, a Figura 5.35(b) com o EKF\_2 e a Figura 5.35(c) o PF.



(a) Modelo EKF\_1.



(b) Modelo EKF\_2.



(c) Modelo PF.

Figura 5.35: Teste em simulação do tipo oito, não utilizando a IMU.

Os principais parâmetros do estado são apresentados em gráfico na Figura 5.36, na Figura 5.37, a Figura 5.38 e na Figura 5.39 correspondentes a  $x_{k+1}$ ,  $y_{k+1}$ ,  $\theta_{k+1}$  e  $v_{k+1}$ , respectivamente.

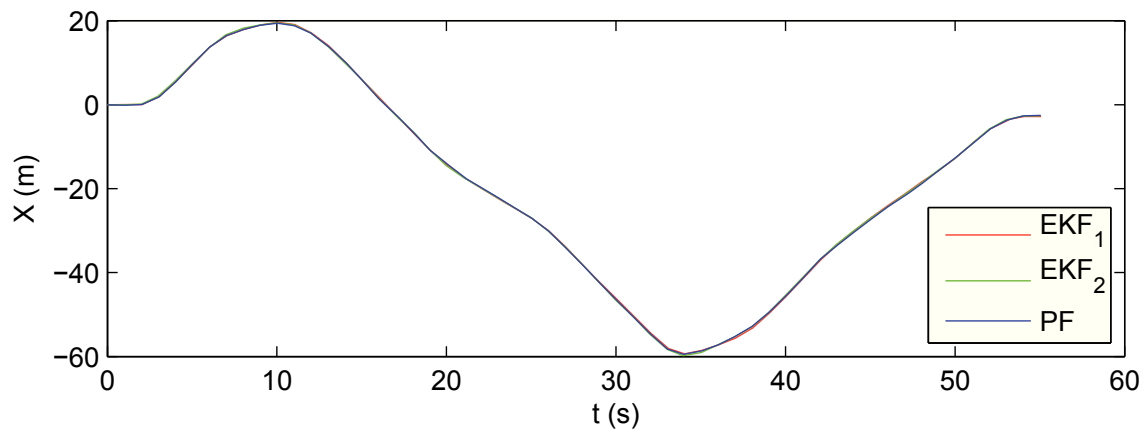


Figura 5.36: *Evolução da posição em X ao longo do percurso tipo oito.*

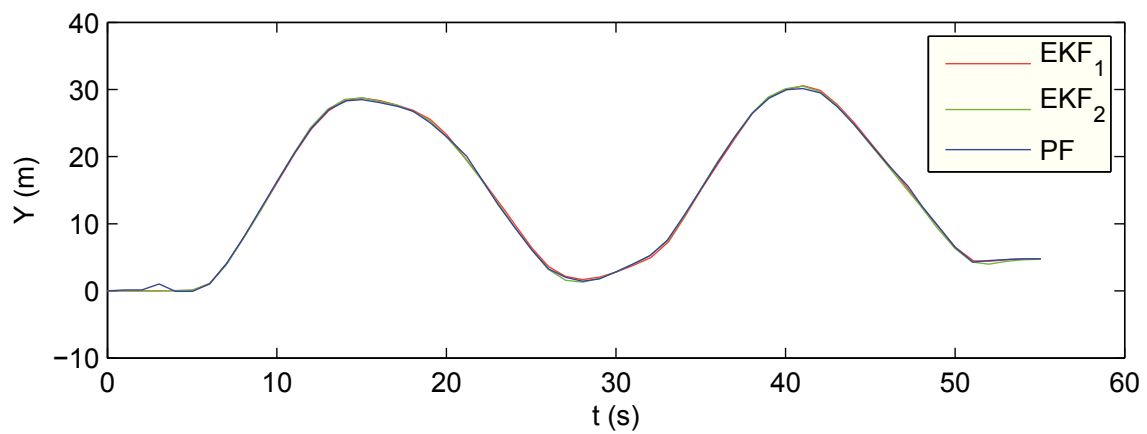


Figura 5.37: *Evolução da posição em Y ao longo do percurso tipo oito.*

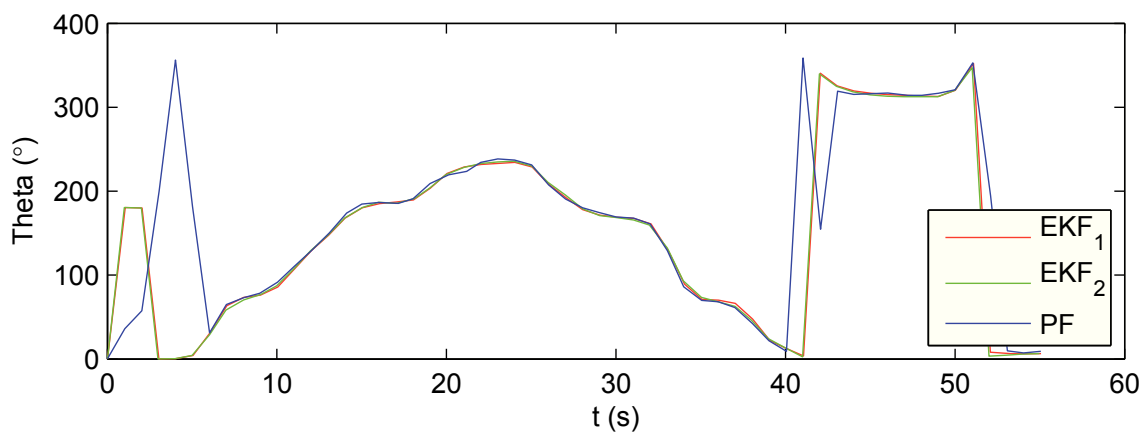


Figura 5.38: *Evolução da orientação ao longo do percurso tipo oito.*

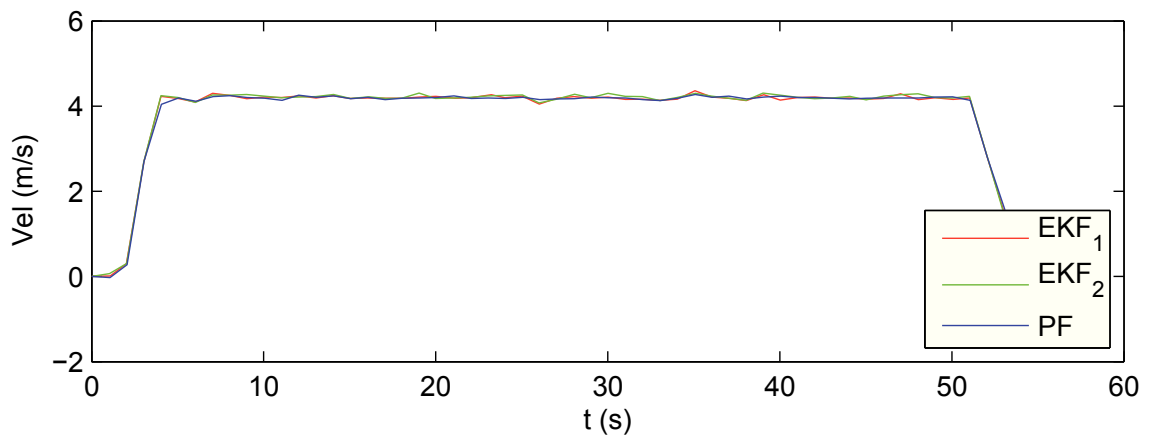


Figura 5.39: *Evolução de velocidade linear ao longo do percurso tipo oito.*

É possível verificar que ambos os métodos tiveram resultados aceitáveis e bastante próximos. Para que seja possível fazer a comparação, é apresentado o erro obtido em termos de posição, orientação e velocidade linear na Figura 5.40, Figura 5.41 e na Figura 5.42, respetivamente.

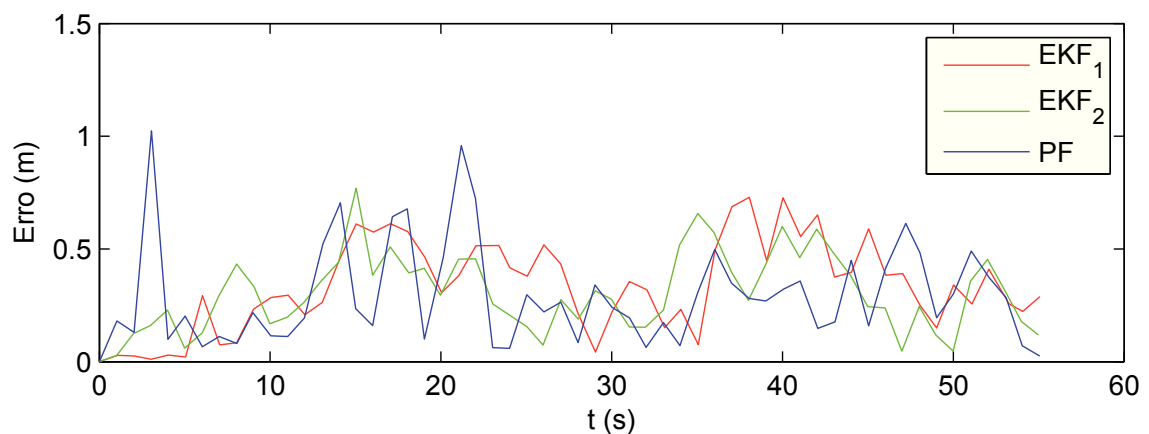


Figura 5.40: *Erro de posição ao longo do percurso tipo oito.*

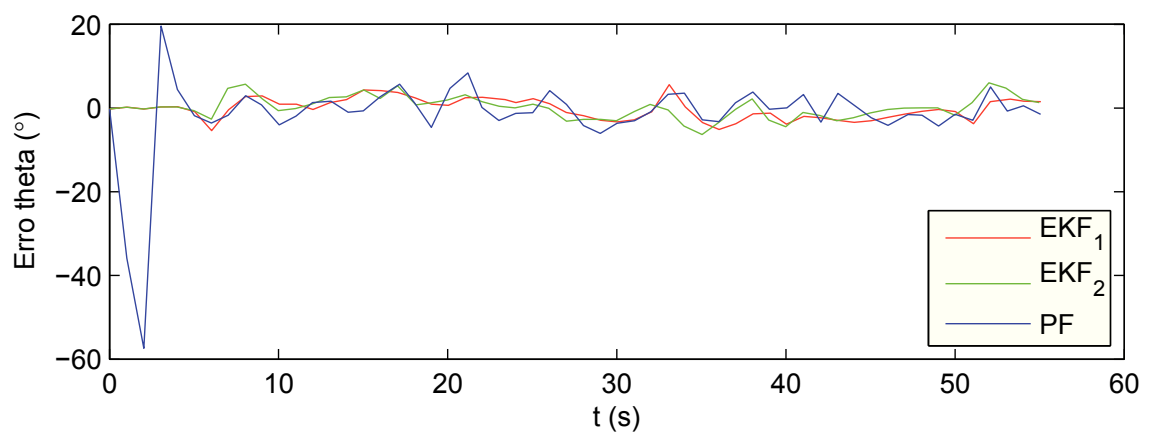


Figura 5.41: *Erro de orientação ao longo do percurso tipo oito.*

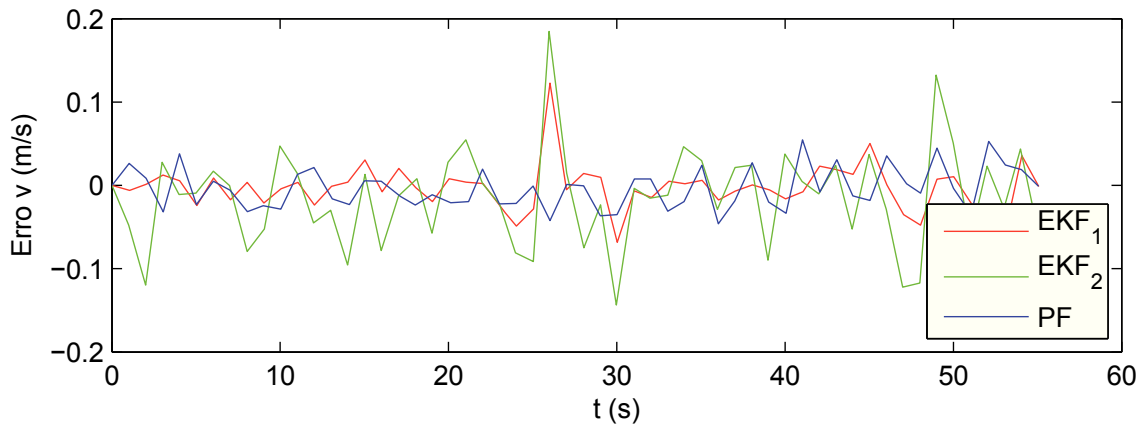


Figura 5.42: Erro de velocidade ao longo do percurso tipo oito.

Na Figura 5.43 é calculado o erro em posição assumindo que o GPS não tem erro associado, ou seja, o GPS é considerado como referência, ficando-se assim com uma melhor ideia do erro associado ao filtro e do erro associado ao GPS. No caso do erro da orientação, assumindo um valor intermédio, é possível perceber que o erro é praticamente constante, sofrendo apenas alterações quando são feitas alterações mais bruscas de orientação, comparando com a Figura 5.29. A velocidade linear demonstrou ter tempos de estabilização diferentes para cada método, sendo bastante linear e semelhante depois de estabilizado.

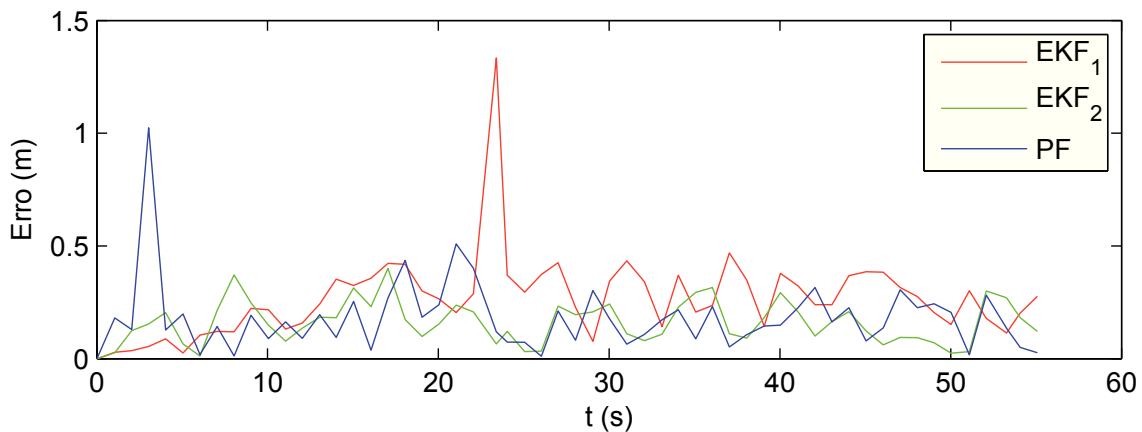


Figura 5.43: Erro de distância ao longo do percurso tipo oito tendo o GPS como referência.

### Comparação para o percurso tipo oito em simulação:

Na Tabela 5.5 está representado o desvio padrão e a média do erro, obtidos para os métodos em estudo, no percurso do tipo oito, fazendo a distinção entre a utilização e a não utilização da IMU. De novo, a vermelho está representado o método que apresentou melhores resultados em termos de posição, a verde em termos de orientação e a azul de velocidade linear.

Para um percurso de natureza diferente, a IMU demonstrou, no teste do tipo oito em

Tabela 5.5: *Comparação entre algoritmos na simulação para o percurso do tipo oito.*

	Método	Utilizando IMU		Não utilizando IMU		Un
		Desvio Padrão	Erro médio	Desvio Padrão	Erro médio	
Posição	EKF_1	0.24712	0.41105	0.20512	0.33777	m
	EKF_2	0.24278	0.35527	0.19273	0.30747	
	PF	0.56928	0.76605	0.28441	0.32684	
Orientação	EKF_1	2.84000	-0.29354	2.48503	-0.17858	° (grau)
	EKF_2	3.57120	0.20786	2.91260	0.16686	
	PF	9.30301	-1.04080	11.59590	-2.17374	
Velocidade linear	EKF_1	0.12490	-0.00826	0.12305	-0.01435	m/s
	EKF_2	0.13070	-0.02254	0.14251	-0.02563	
	PF	0.12787	-0.00310	0.11995	-0.01498	

ambiente de simulação, não ser uma boa opção para a estimação da posição e orientação e, ao contrário de no teste em elipse, esta trouxe ligeiro melhoramento em termos de velocidade. O EKF\_2 sem utilização da IMU demonstrou ser o método com melhores resultados em posição e orientação, tendo o menor erro em média. Na velocidade linear, o método que obteve melhor resultado foi o PF com utilização da IMU, também com o menor erro em média.

Após estes testes, pode ser feita uma comparação mais conclusiva para os métodos implementados em simulação. A Tabela 5.6 apresenta os resultados obtidos para cada algoritmo, nos dois testes em ambiente de simulação, a fim de saber qual o melhor em geral.

Tabela 5.6: *Comparação entre algoritmos para todos os testes efetuados em simulação.*

	Método	Utilizando IMU		Não utilizando IMU		Un
		Desvio Padrão	Erro médio	Desvio Padrão	Erro médio	
Posição	EKF_1	0.30989	0.50116	0.25695	0.46561	m
	EKF_2	0.26643	0.45388	0.23407	0.42932	
	PF	0.49310	0.70124	0.37675	0.49129	
Orientação	EKF_1	2.56468	1.06205	2.02046	0.62722	° (grau)
	EKF_2	2.62819	0.84093	2.19466	0.54218	
	PF	6.93558	0.06961	8.68908	-1.41175	
Velocidade linear	EKF_1	0.09094	-0.00735	0.08653	-0.00394	m/s
	EKF_2	0.09376	-0.01735	0.10298	-0.01933	
	PF	0.30431	-0.01861	0.08250	-0.00523	

A partir destes resultados finais, pode-se concluir que a utilização da IMU, em ambiente de simulação, não trás benefícios ao sistema, tendo obtido maior erro em média em quase todos os métodos, aquando a sua utilização.

Ficou também claro que, em ambiente de simulação, o método EKF\_2 sem utilização da IMU foi o que método se obteve melhores resultados em posição e orientação. Em velocidade linear, o método que obteve melhores resultados foi o EKF\_1, com a menor média do erro. Neste trabalho é dada maior importância à posição, em segundo lugar à orientação, e por último à velocidade linear. Assim, com melhor desempenho em posição e na orientação, o EKF\_2 sem utilização da IMU é considerado o método com melhores resultados, para estes teste em ambiente de simulação.

### 5.1.2 Ambiente real

Este ambiente tem a restrição de ser limitado ao estacionamento da Escola Superior de Tecnologia e Gestão (ESTG) de Leiria, visto que o veículo não pode circular em via pública. Assim, os testes realizados nos parques de estacionamento foram restritos ao espaço disponível. Na Figura 5.44 estão representados, em vista satélite, os locais utilizados para os testes, sendo eles o local para o teste do tipo elipse e do tipo oito, e o outro local para o teste de utilização normal.

Procurou-se que os testes em ambiente real fossem semelhantes aos testes em ambiente de simulação, tendo em consideração as limitações existentes. Ao contrário do ambiente de simulação, a velocidade pretendida não é imposta a um controlador de velocidade, mas controlada manualmente, não conseguindo por isso ter uma velocidade tão estável.

Neste ambiente não havia acesso a medidas reais, tendo-se por isso assumido o GPS e a odometria como referência. O GPS, foi utilizado como referência da posição e da orientação, por ser o único sensor absoluto e ter o menor erro em média, nestes parâmetros. A odometria, foi utilizada como referencia em velocidade, por ser o sensor com menor erro, em média, de velocidade.

#### Teste tipo elipse

Este teste foi realizado a uma velocidade linear um pouco superior, pois em condução manual, quanto mais baixa for a velocidade pretendida, mais difícil o seu controlo. Assim, o teste da elipse foi realizado a uma velocidade linear em torno dos 4.5 m/s e teve como resultante uma elipse com uma largura de aproximadamente 100 m.

#### - Teste tipo elipse utilizando Odometria, GPS e IMU

Na Figura 5.45 é feita a comparação do percurso estimado, pelos diferentes métodos em análise neste trabalho, para o percurso do tipo elipse. A Figura 5.45(a) representa o teste realizado utilizando o método EKF\_1, a Figura 5.45(b) representa o método EKF\_2, a Figura 5.5(c) o método PF e a Figura 5.45(d) ambos os modelos sobrepostos em mapa.



Figura 5.44: Vista satélite da ESTG, via Google<sup>®</sup>.

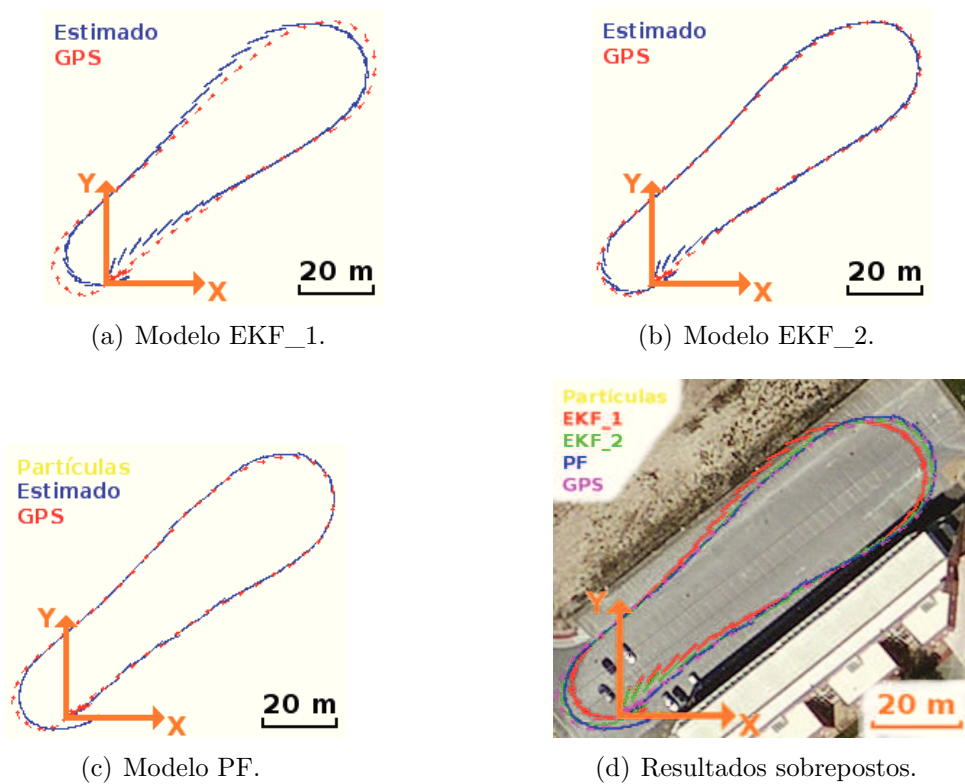


Figura 5.45: Teste em ambiente real do tipo elipse, utilizando todos os sensores.

Mais uma vez, a trajetória estimada pelo método em estudo é representada a azul; a trajetória adquirida pelo GPS a vermelho; e a cor amarela representa as várias partículas estimadas, no caso do filtro de partículas. Na Figura dos resultados sobrepostos, Figura 5.45(d) o GPS está representado a magenta, o EKF\_1 a vermelho, o EKF\_2 a verde e o PF a azul. Nesta figura, o percurso foi deslocado de modo a que o percurso efetuado correspondesse com o demonstrado, caso contrário estaria ligeiramente desviado. Este desvio deve-se principalmente ao erro associado ao GPS, ver Secção A.2.2, dado que não estamos a utilizar um GPS de precisão nem qualquer estação de base.

Para melhor comparação dos métodos, são representados em gráfico os parâmetros do estado. Na Figura 5.46, Figura 5.47, a Figura 5.48 e a 5.49, referente a  $x_{k+1}$ ,  $y_{k+1}$ ,  $\theta_{k+1}$  e  $v_{k+1}$ , respetivamente, de cada método.

Ao contrário dos testes em simulação, aqui já é possível observar algumas diferenças entre os métodos. Voltando a referir que o GPS foi considerado como referência para a localização e orientação, e a odometria considerada referência da velocidade linear, na Figura 5.50, Figura 5.51 e na Figura 5.52 é apresentado o erro obtido em posição, orientação e velocidade linear, respetivamente.

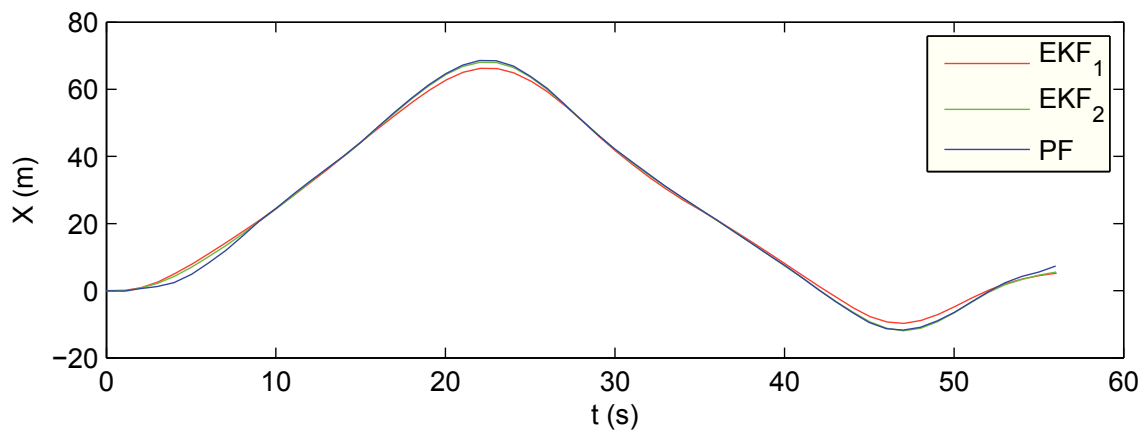


Figura 5.46: *Evolução da posição em X ao longo do percurso tipo elipse.*

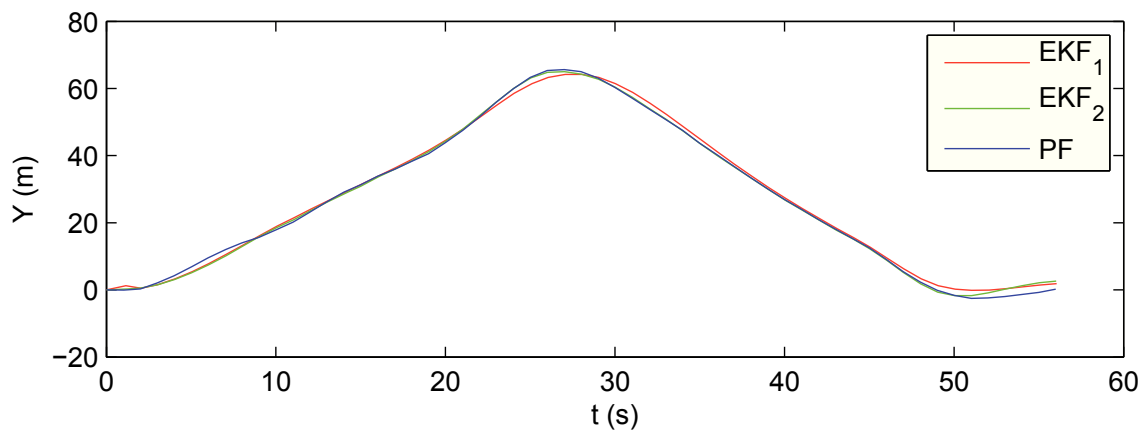


Figura 5.47: *Evolução da posição em Y ao longo do percurso tipo elipse.*

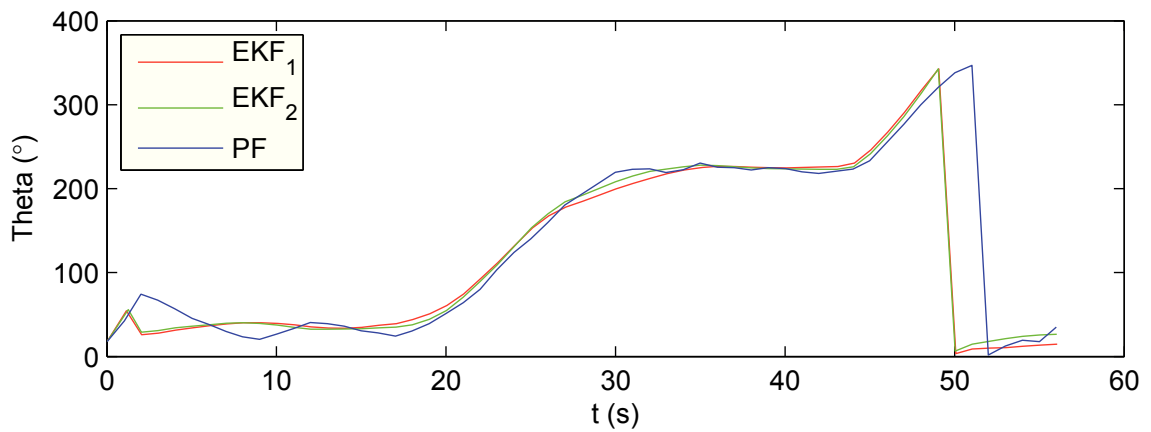


Figura 5.48: *Evolução da orientação ao longo do percurso tipo elipse.*

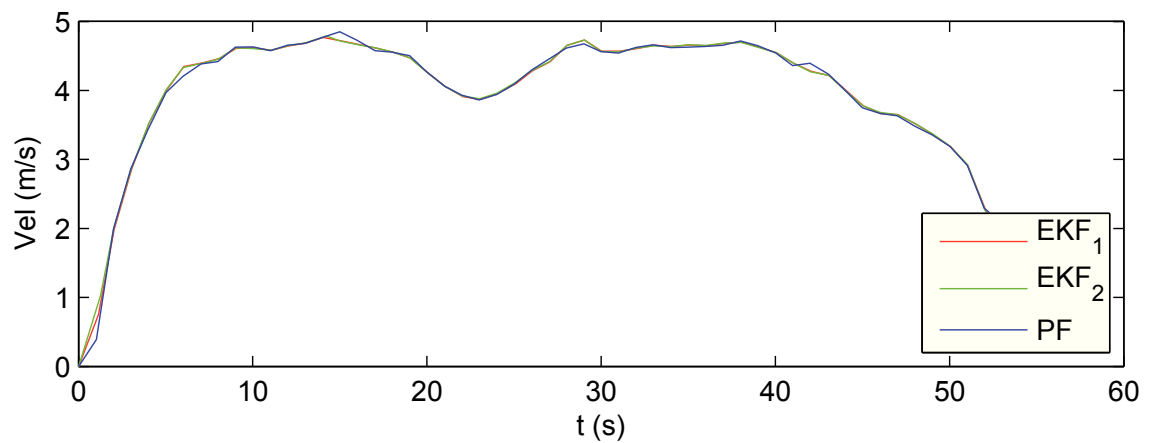


Figura 5.49: *Evolução da velocidade linear ao longo do percurso tipo elipse.*

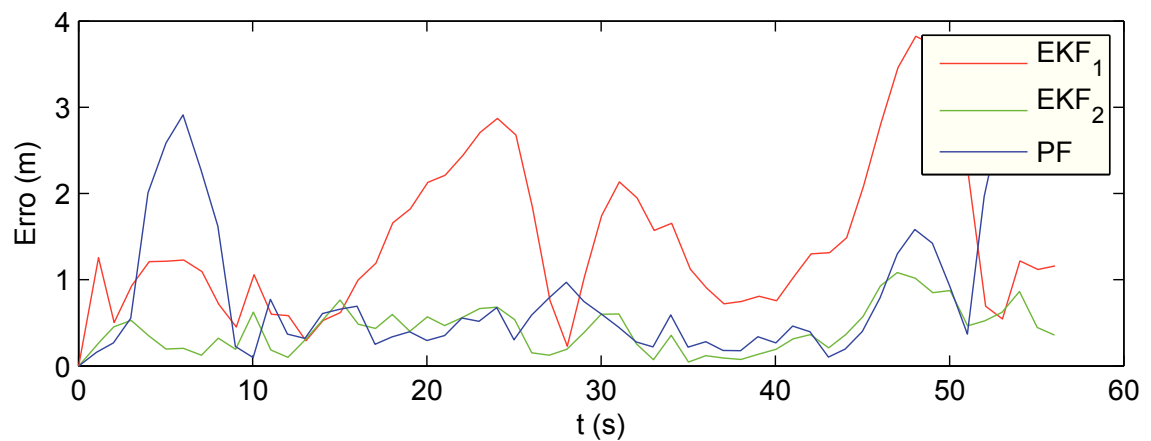


Figura 5.50: *Erro de posição ao longo do percurso tipo elipse em ambiente real.*

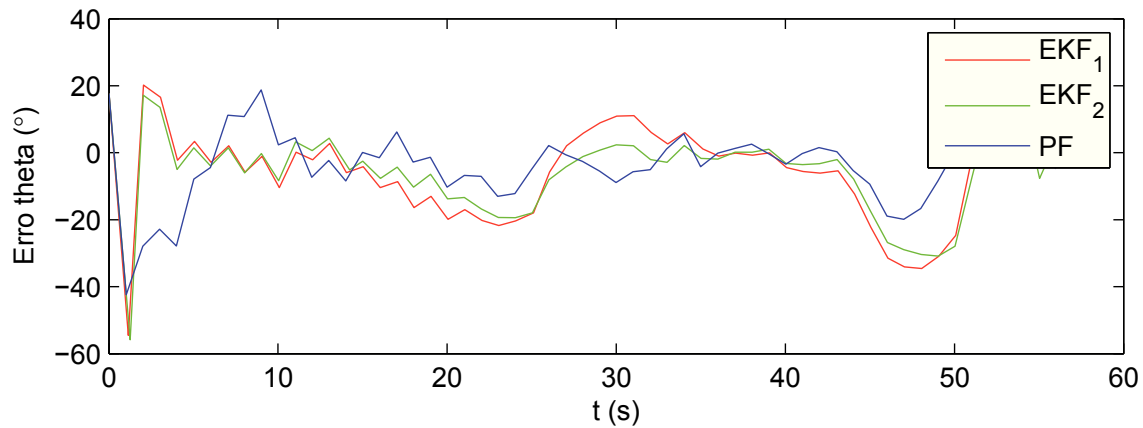


Figura 5.51: Erro de orientação ao longo do percurso tipo elipse em ambiente real.

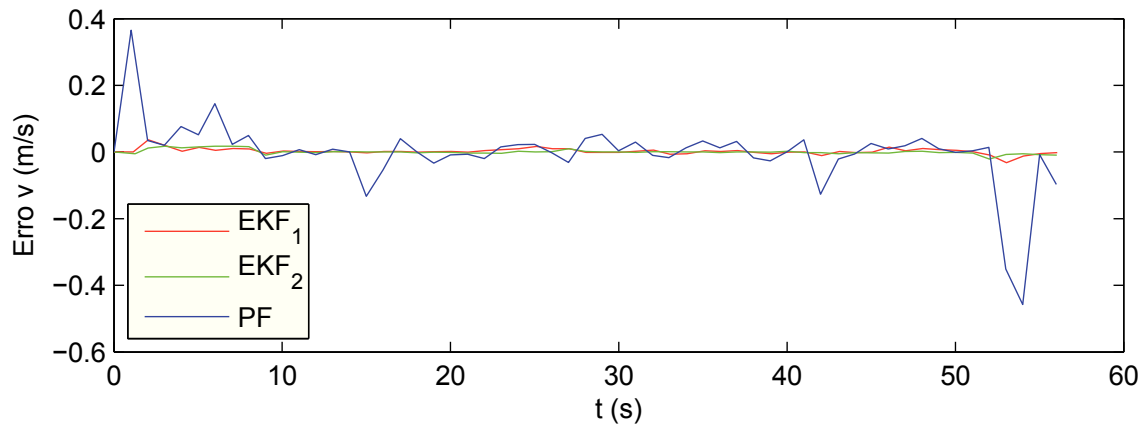


Figura 5.52: Erro de velocidade ao longo do percurso tipo elipse em ambiente real.

No erro de posição é claramente visível que o EKF\_1 obteve um resultado bastante irregular e com erros grandes, atingindo quase 4 m de erro. Fora o tempo superior de convergência inicial, o PF e o EKF\_2 obtiveram resultados bastante semelhantes em posição. Na orientação, o PF demonstra outra vez ter um tempo de convergência inicial superior, seguindo-se de resultados semelhantes com os outros métodos. Comparando com a Figura 5.48, é possível verificar que o erro em orientação aumenta quanto maior for a variação de orientação.

Mesmo sendo a medida mais realista, a odometria não é a referência ideal para calcular o erro, visto a sua grande influência sobre o sistema, originada pela importância associada. Isto é demonstrado no sentido em que o EKF\_1 e o EKF\_2 têm erro médio quase nulo.

Na Tabela 5.7 é feita a comparação dos métodos em estudo em termos de posição, orientação e velocidade linear, para o percurso do tipo elipse em ambiente real.

#### - Teste tipo elipse utilizando Odometria e GPS, não utilizando a IMU

A Figura 5.53 representa o mesmo teste que o anterior, mas não utilizando a IMU. Nesta comparação, a Figura 5.53(a) representa o teste realizado utilizando o método

EKF\_1, a Figura 5.53(b) representa o método EKF\_2, a Figura 5.53(c) o método PF e a Figura 5.53(d) ambos os modelos sobrepostos em mapa.

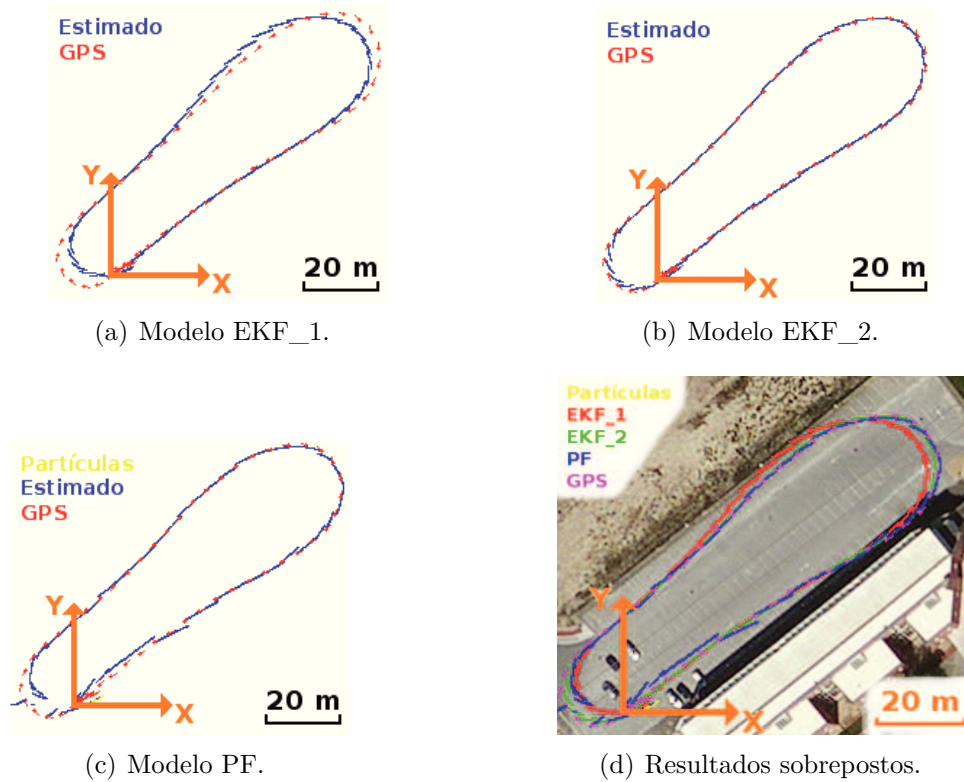


Figura 5.53: *Teste em ambiente real do tipo elipse, não utilizando a IMU.*

De novo são apresentados os principais parâmetros do estado em gráfico, sendo a Figura 5.54, a Figura 5.55, a Figura 5.56 e a Figura 5.57 correspondentes a  $x_{k+1}$ ,  $y_{k+1}$ ,  $\theta_{k+1}$  e  $v_{k+1}$ , respectivamente.

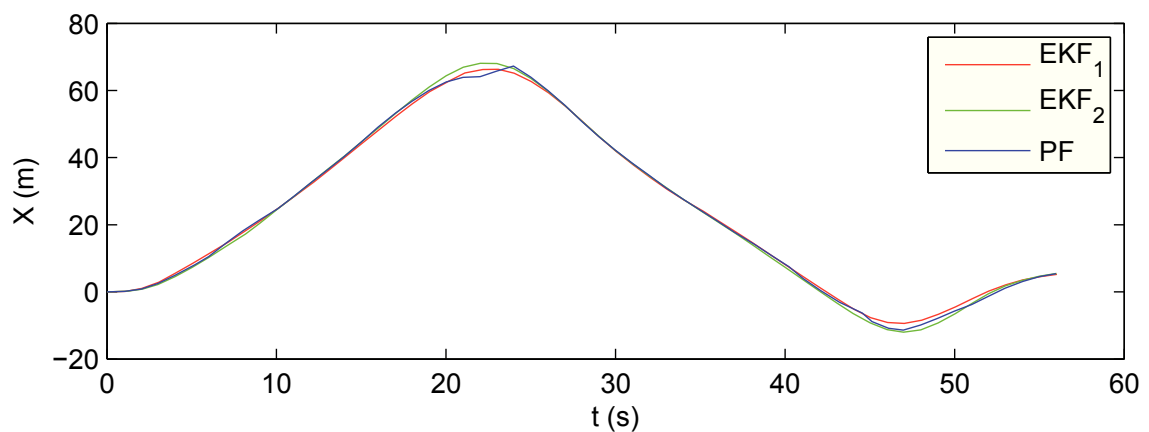


Figura 5.54: *Evolução da posição em X ao longo do percurso tipo elipse.*

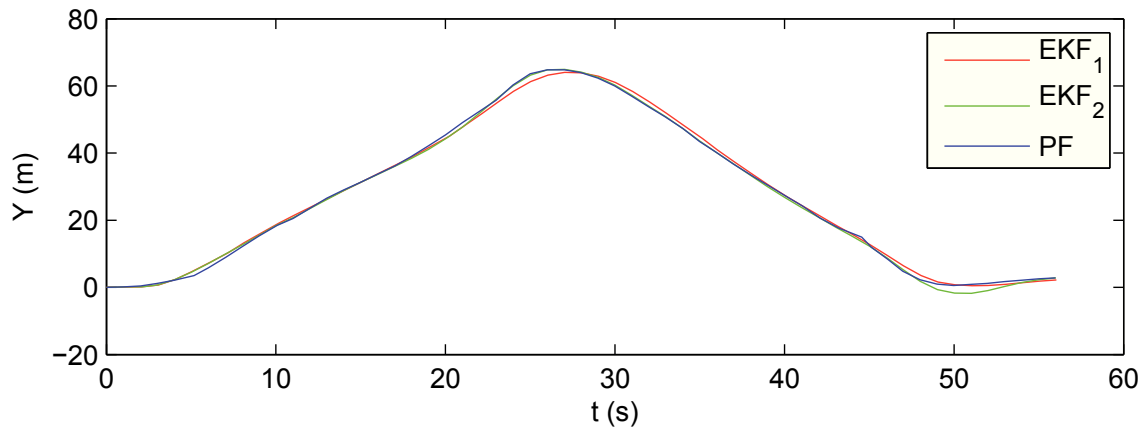


Figura 5.55: *Evolução da posição em Y ao longo do percurso tipo elipse.*

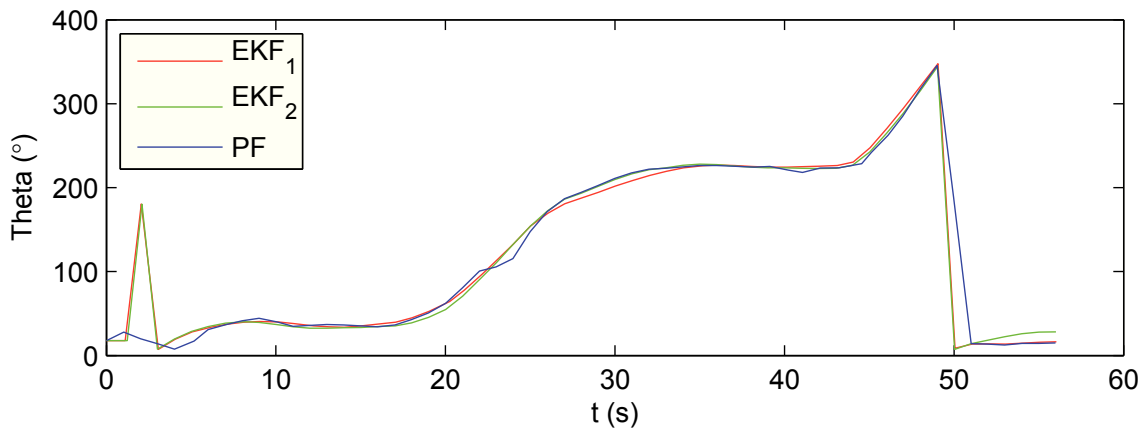


Figura 5.56: *Evolução da orientação ao longo do percurso tipo elipse.*

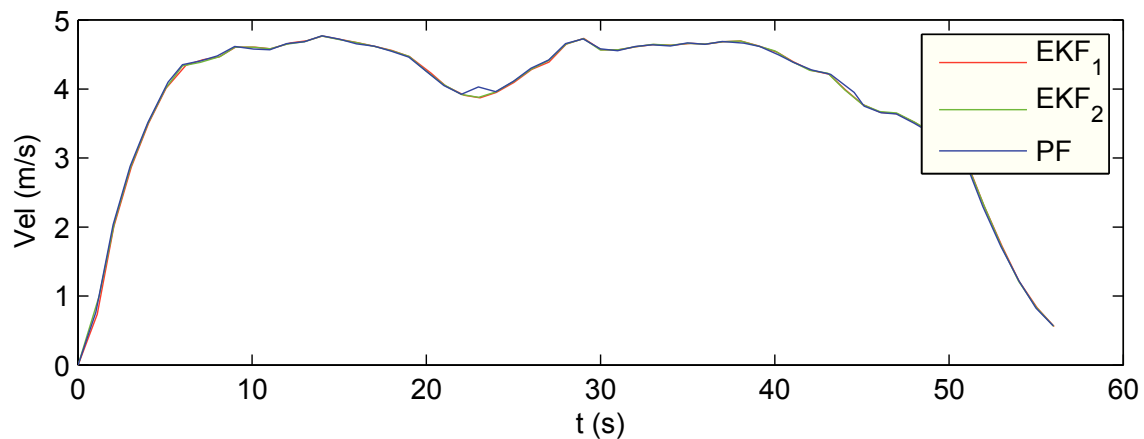


Figura 5.57: *Evolução da velocidade ao longo do percurso tipo elipse.*

Nestes resultados, os parâmetros obtidos entre métodos têm maior semelhança entre eles do que os obtidos no método onde foi utilizado a IMU, mas não ao ponto de ficarem totalmente sobrepostos, como acontece em ambiente de simulação.

Na Figura 5.58 e na Figura 5.59 é apresentado o erro relativo à posição e à orientação, respetivamente, quando comparado com o GPS, e na Figura 5.60 é apresentado o erro relativo à velocidade linear, quando comparado com a odometria.

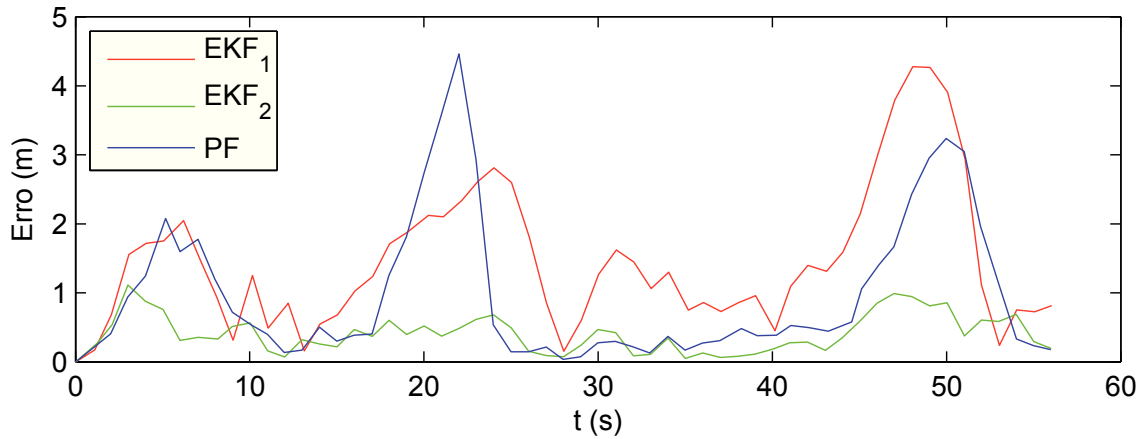


Figura 5.58: Erro de posição ao longo do percurso tipo elipse.

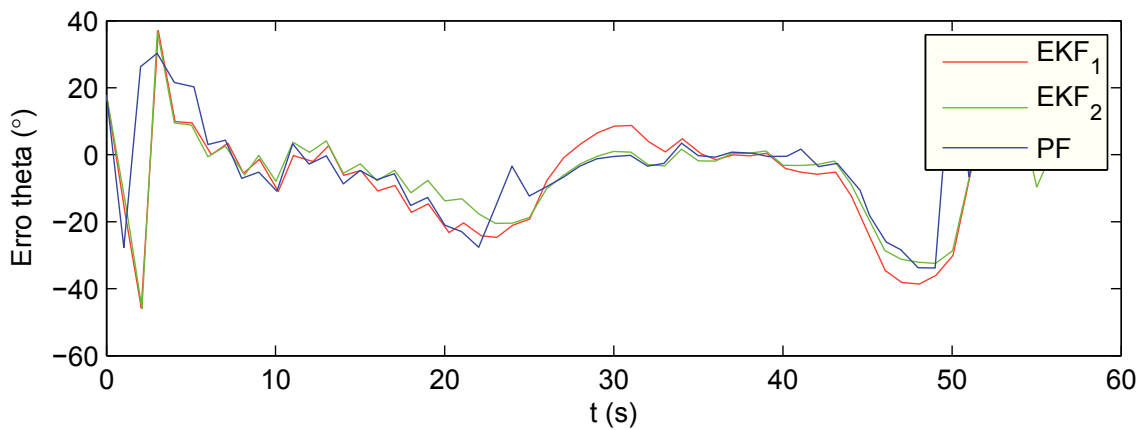


Figura 5.59: Erro de orientação ao longo do percurso tipo elipse.

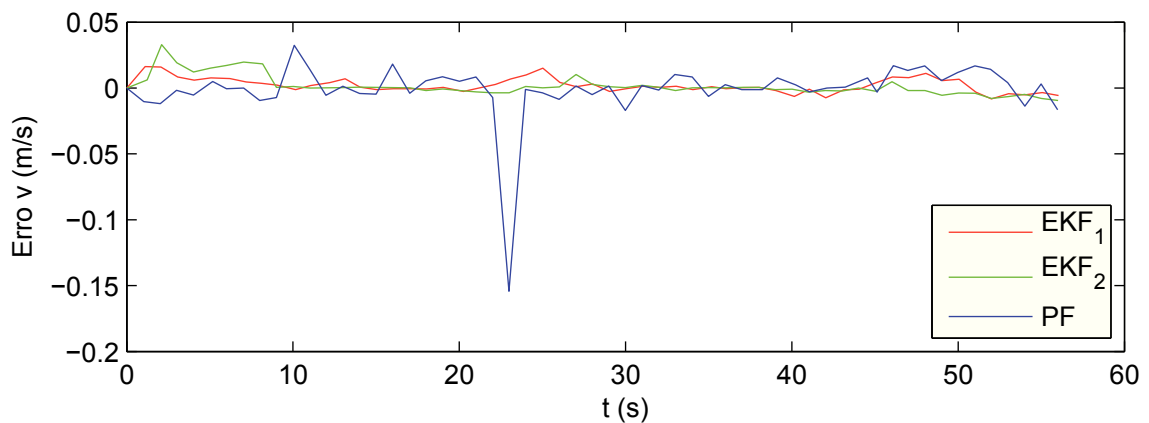


Figura 5.60: Erro de velocidade ao longo do percurso tipo elipse.

### Comparação para o percurso tipo elipse em ambiente real:

Na Tabela 5.7 está representado o desvio padrão e a média do erro, obtidos para os métodos em estudo, no percurso da elipse em ambiente real, fazendo a distinção entre a utilização e a não utilização da IMU.

Tabela 5.7: Comparação entre algoritmos em ambiente real para o percurso do tipo elipse.

	Método	Utilizando IMU		Não utilizando IMU		Un
		Desvio Padrão	Erro médio	Desvio Padrão	Erro médio	
Posição	EKF_1	0.95356	1.44784	1.06637	1.46347	$m$
	EKF_2	0.37295	0.42715	0.40302	0.41328	
	PF	0.94091	0.90303	1.12660	1.02664	
Orientação	EKF_1	16.21360	-4.76030	17.14350	-4.33157	° (grau)
	EKF_2	13.65891	-5.74113	14.65963	-4.26967	
	PF	12.72610	-2.76709	15.42906	-3.90508	
Velocidade linear	EKF_1	0.01027	0.00220	0.00570	0.00205	$m/s$
	EKF_2	0.00981	0.00050	0.01088	0.00153	
	PF	0.11298	-0.00366	0.03178	-0.00109	

Onde a vermelho está representado o método que apresentou melhores resultados de posição, a verde de orientação e a azul de velocidade linear.

É importante voltar a referir que estas comparações são feitas considerando que a localização e orientação do GPS e a velocidade da odometria como referências, assumindo que não têm erro, o que na prática não se verifica, e foram utilizadas como tal por serem a melhor opção disponível.

À semelhança do teste em elipse no ambiente de simulação, o EKF\_2 sem a utilização da IMU foi o método que obteve melhores resultados em posição, tendo obtido um menor erro em média. Na orientação, com um erro médio significativamente mais baixo, o PF com utilização da IMU foi o melhor resultado, enquanto que em velocidade foi o EKF\_2 com utilização da IMU. Pode-se concluir que o EKF\_2 sem utilização da IMU foi o filtro que obteve melhores resultados, dando prioridade à posição perante a diferença obtida na orientação e na velocidade linear.

### Teste tipo oito

Como no teste do tipo elipse realizado em ambiente real, o teste foi realizado a uma velocidade linear em torno dos 4.5 m/s e teve como resultante um oito com uma largura de aproximadamente 100 m.

- Teste tipo oito utilizando Odometria, GPS e IMU

A Figura 5.61(a) representa o teste realizado utilizando o EKF\_1, a Figura 5.61(b) com o EKF\_2, a Figura 5.61(c) o PF e a Figura 5.61(d) estão representados ambos os testes sobrepostos em mapa.

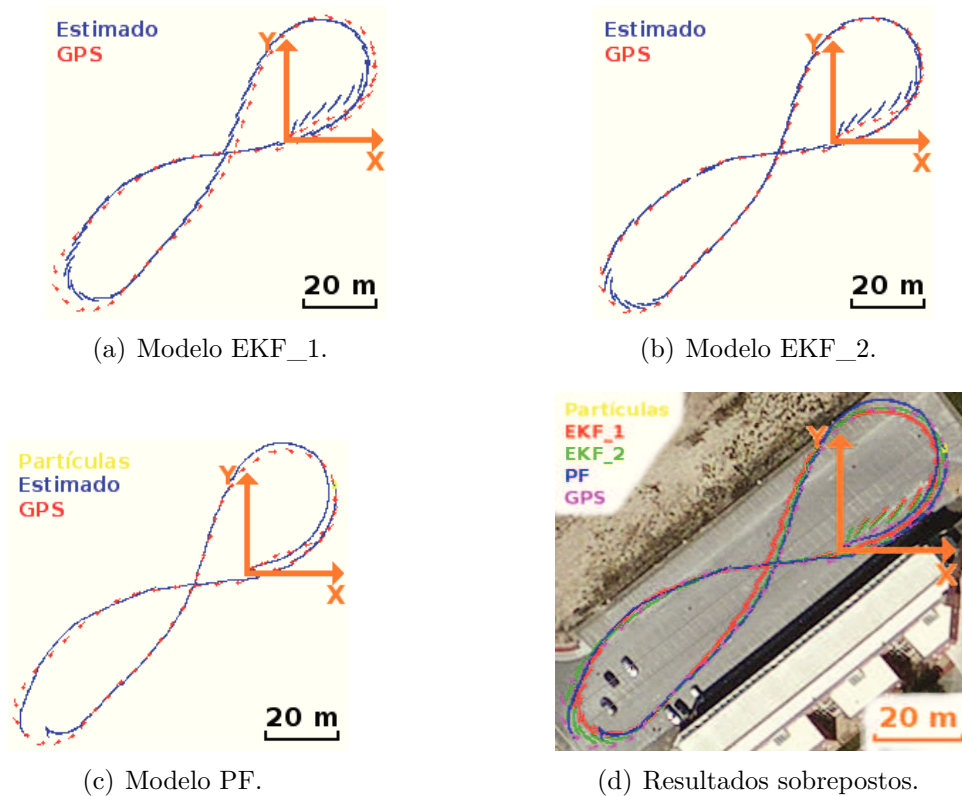


Figura 5.61: *Teste em ambiente real do tipo oito.*

Nestes resultados, é possível observar que, inicialmente, no método EKF\_1 e no método EKF\_2, o estado tem alguma divergência. Como previsto, esta divergência é atenuada ao longo do tempo, e é devida à orientação inicial do GPS estar errada. No modelo PF esta situação não tem tanta influência, visto existirem diversas estimativas do estado, o que origina que o modelo tenda a convergir mais rapidamente para a solução correta. Pelos mesmos motivos do teste tipo elipse (em ambiente real), mais uma vez, na figura com os resultados sobrepostos no mapa (Figura 5.61(d)), o percurso foi deslocado de modo a que o percurso efetuado correspondesse com o demonstrado.

Os principais parâmetros do estado estão representados na Figura 5.62, Figura 5.63, na Figura 5.64 e na Figura 5.65, correspondentes a  $x_{k+1}$ ,  $y_{k+1}$ ,  $\theta_{k+1}$  e  $v_{k+1}$ , respetivamente, de cada método.

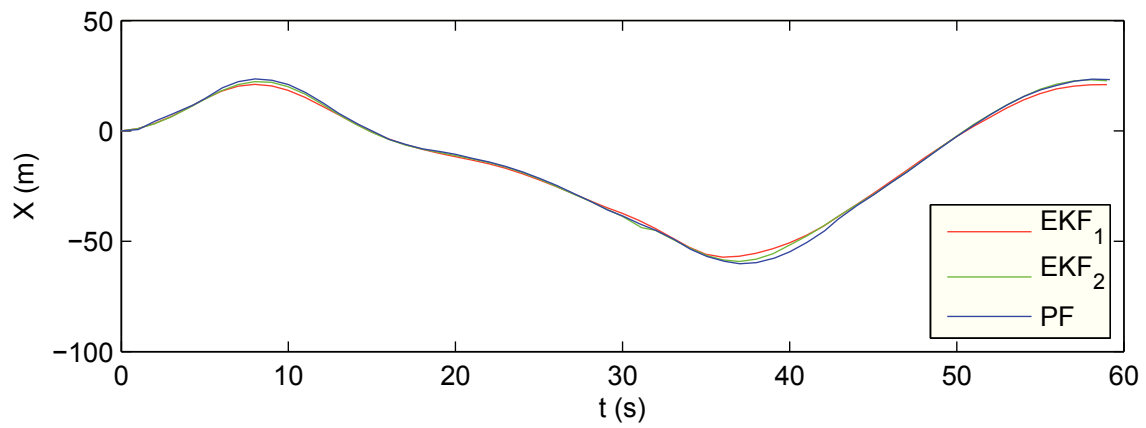


Figura 5.62: *Evolução da posição em X ao longo do percurso tipo oito.*

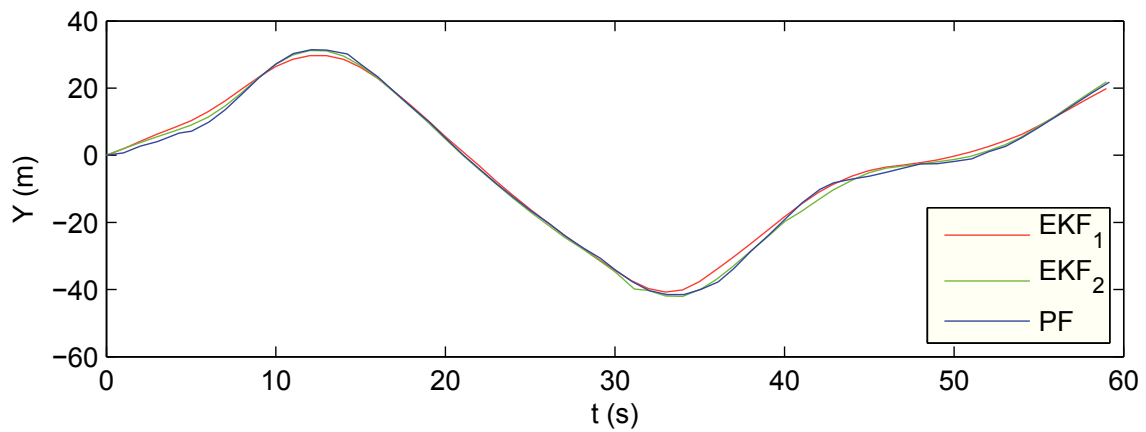


Figura 5.63: *Evolução da posição em Y ao longo do percurso tipo oito.*

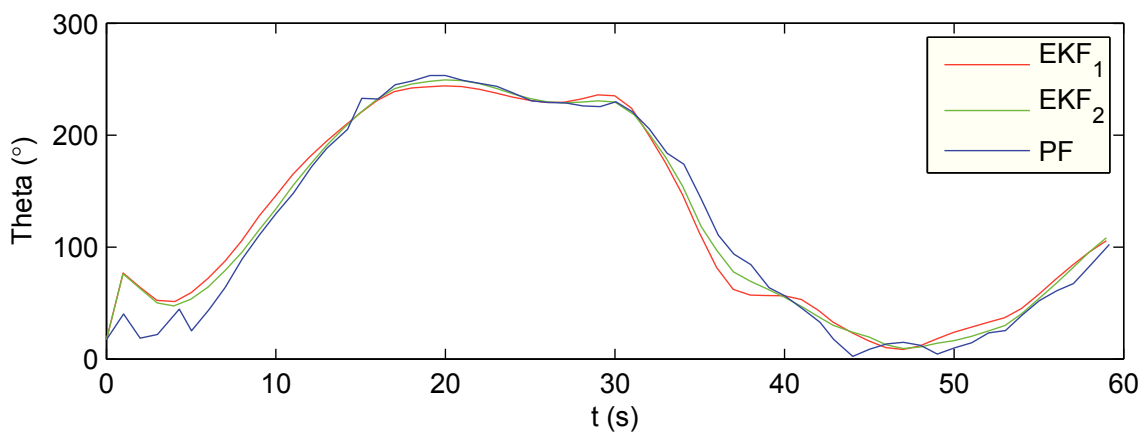


Figura 5.64: *Evolução da orientação ao longo do percurso tipo oito.*

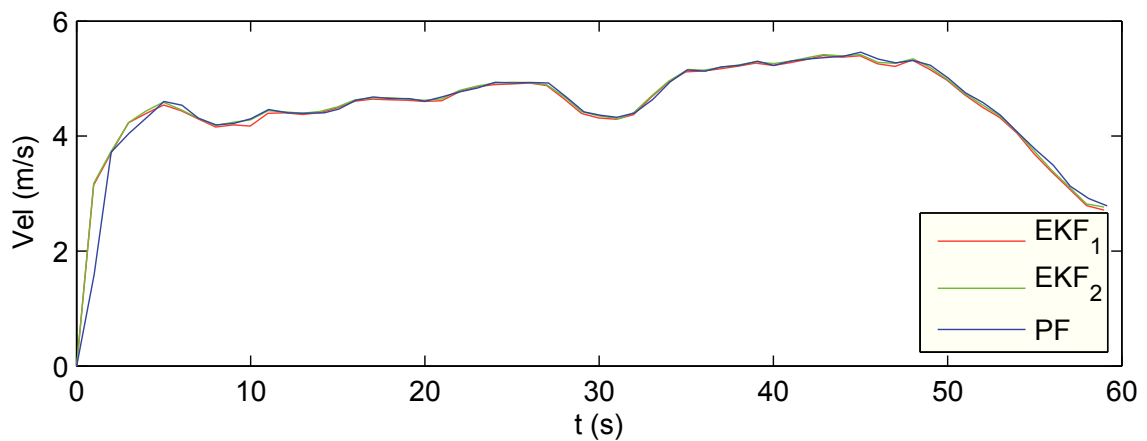


Figura 5.65: *Evolução da velocidade ao longo do percurso tipo oito.*

Na Figura 5.66 e na Figura 5.67 estão representadas as evoluções do erro da posição e da orientação, respetivamente, ao longo do tempo para o mesmo percurso, quando comparado com o GPS. Na Figura 5.68 estão representadas as evoluções do erro da velocidade linear, quando comparado com a odometria.

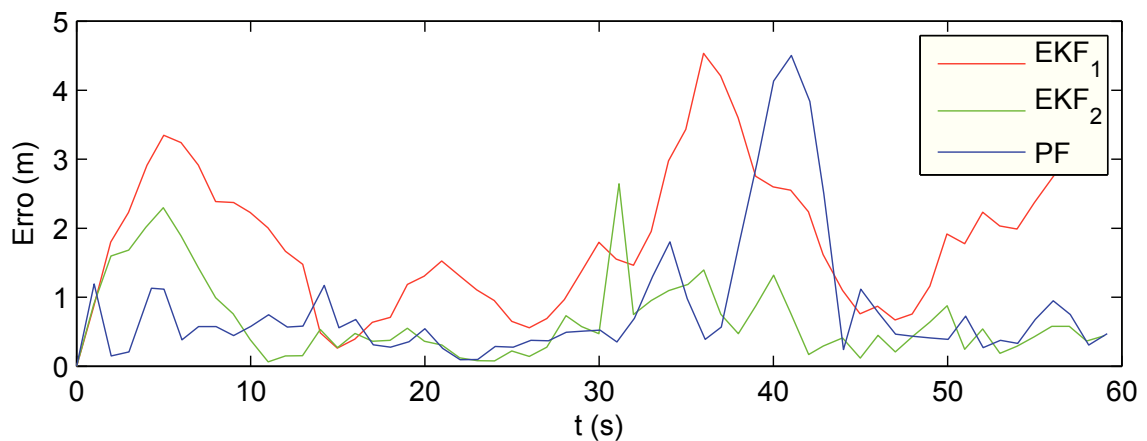


Figura 5.66: *Erro de distância ao longo do percurso tipo oito.*

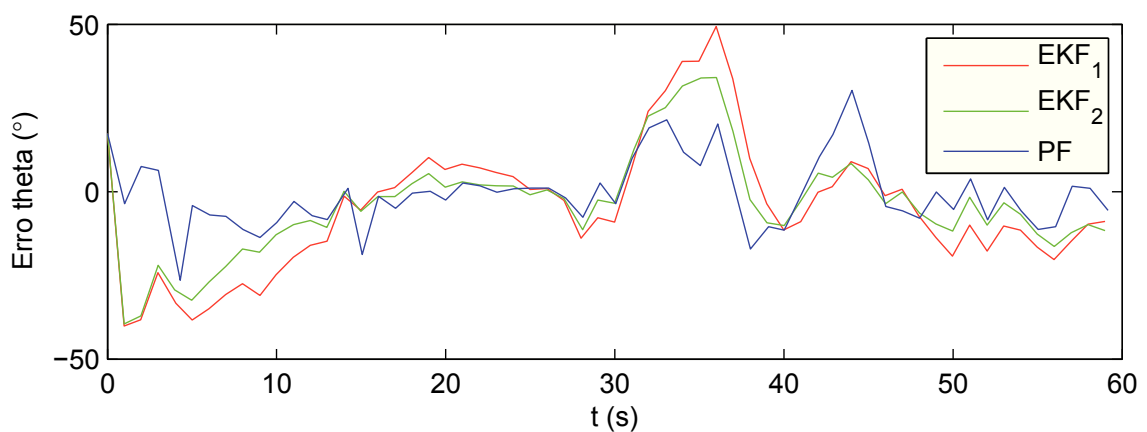


Figura 5.67: *Erro da orientação ao longo do percurso tipo oito.*

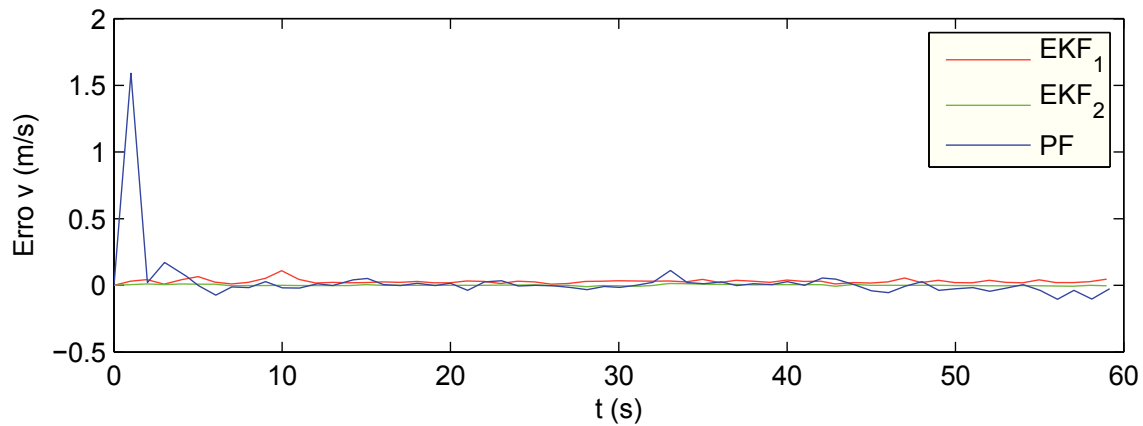


Figura 5.68: Erro da velocidade ao longo do percurso tipo oito.

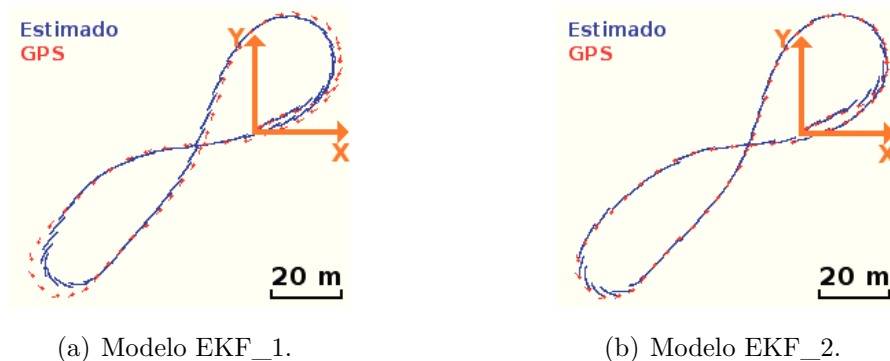
Mais uma vez, o EKF\_1 mostrou ter resultados bastante irregulares e com erros grandes, chegando mesmo a ultrapassar os 4 m de erro. Ainda na posição, o PF demonstrou ter resultados em todo semelhantes com o EKF\_2. Na orientação, o PF continua a ter um tempo de convergência inicial superior, seguindo-se de resultados semelhantes com os outros métodos. Comparando com a Figura 5.64, é possível verificar que o erro em orientação varia consoante a variação de orientação do veículo.

Devido à importância associada à velocidade linear adquirida pela odometria, origina que o erro médio em velocidade seja quase nulo, quando utilizada a odometria como referência.

Na Tabela 5.8 é feita a comparação dos métodos em estudo em termos de posição, orientação e velocidade linear, para o percurso do tipo elipse em ambiente real.

#### - Teste tipo oito utilizando Odometria e GPS, não utilizando a IMU

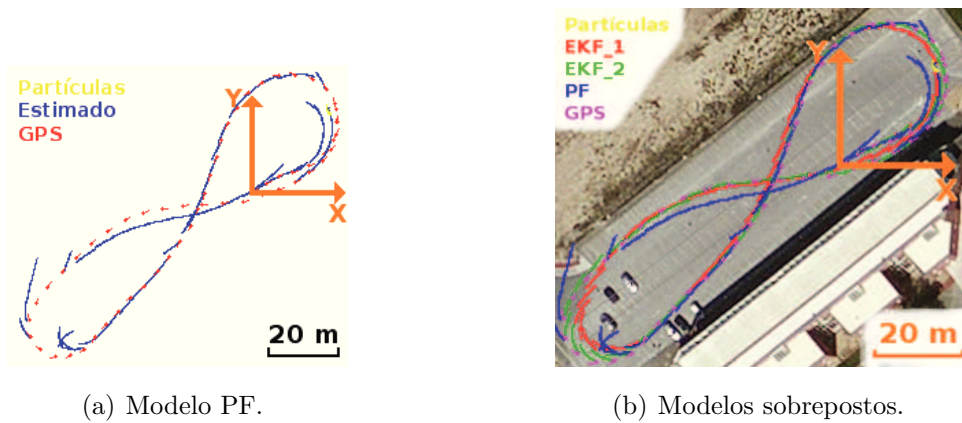
A Figura 5.69(a) representa o teste realizado utilizando o EKF\_1, a Figura 5.69(b) com o EKF\_2, a Figura 5.70(a) o PF e a Figura 5.70(b) ambos os modelos sobrepostos em mapa.



(a) Modelo EKF\_1.

(b) Modelo EKF\_2.

Figura 5.69: Teste em ambiente real do tipo oito.



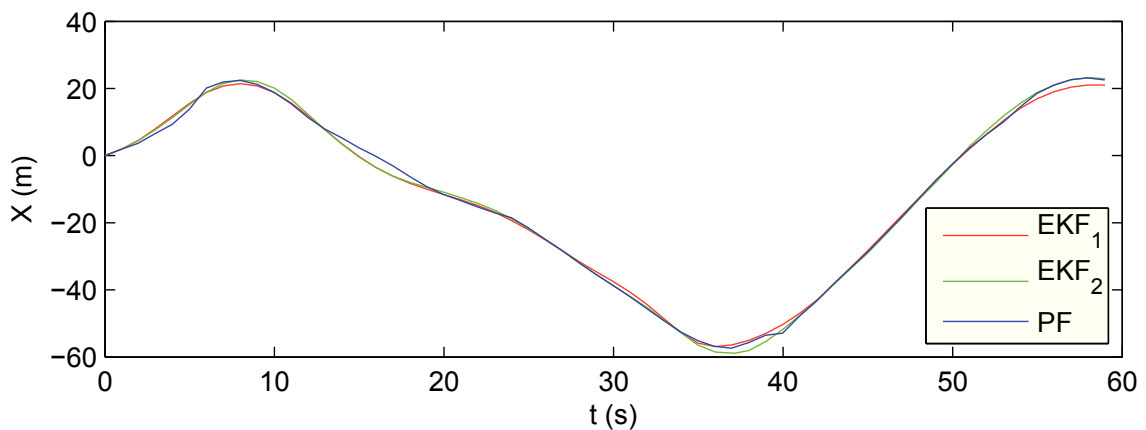
(a) Modelo PF.

(b) Modelos sobrepostos.

Figura 5.70: *Teste em ambiente real do tipo oito.*

À semelhança do teste anterior, teste tipo oito com utilização da IMU, é possível verificar que existem deslocamentos significativos nas curvas, fruto do erro associado ao ângulo da direção, possivelmente reduzido com uma calibração mais rigorosa. Também é mais visível nestes resultados, comparando com os testes anteriores, que quando o sistema está perdido no espaço, no método PF é feita a reinicialização algumas partículas, acelerando conseqüentemente a convergência do estado estimado com o estado medido pelo GPS, ver por exemplo na curva presente no canto inferior esquerdo da Figura 5.70(a).

Na Figura 5.71, Figura 5.72, Figura 5.73 e na Figura 5.74, estão apresentados os principais parâmetros do estado, referentes a  $x_{k+1}$ ,  $y_{k+1}$ ,  $\theta_{k+1}$  e  $v_{k+1}$ , respetivamente, de cada método.

Figura 5.71: *Evolução da posição em X ao longo do percurso tipo oito.*

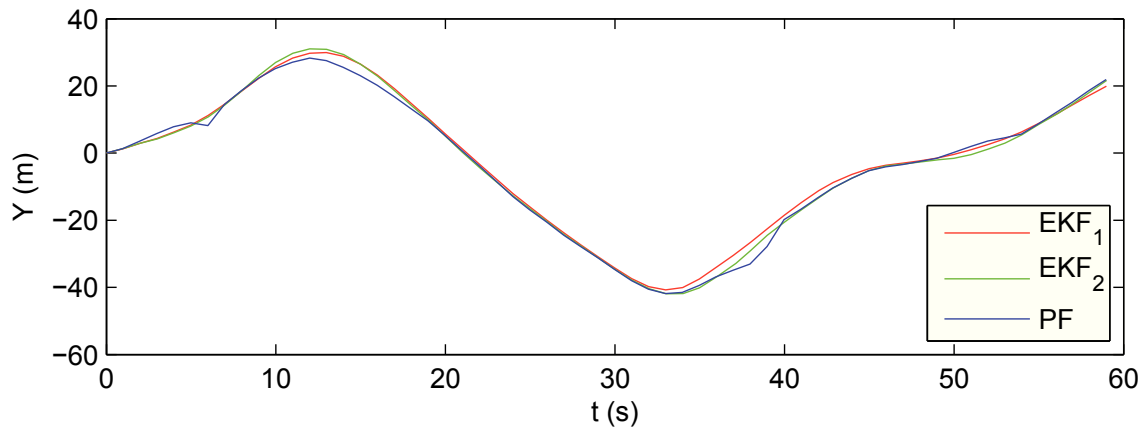


Figura 5.72: *Evolução da posição em Y ao longo do percurso tipo oito.*

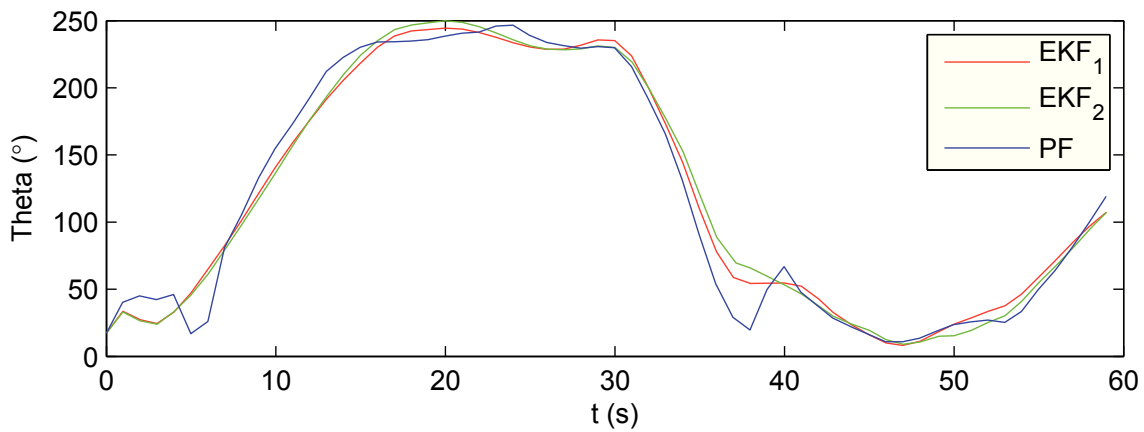


Figura 5.73: *Evolução da orientação ao longo do percurso tipo oito.*

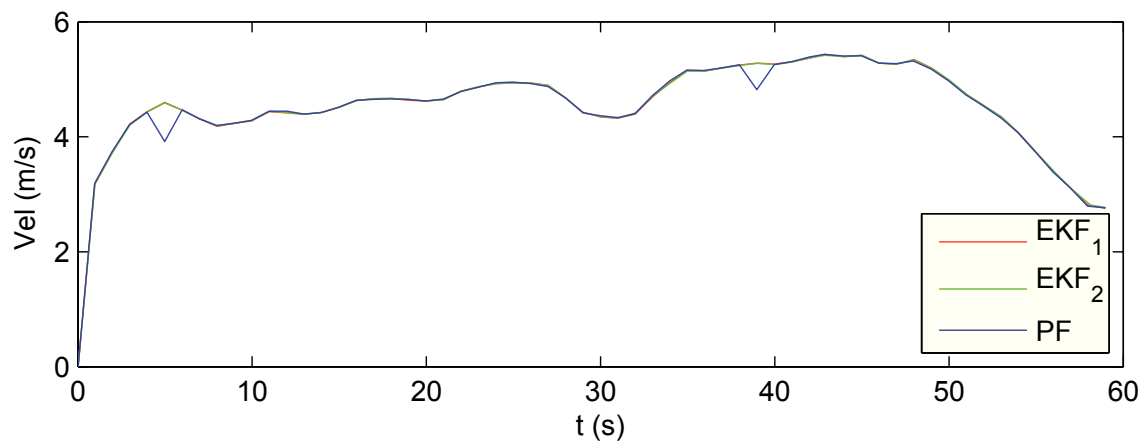


Figura 5.74: *Evolução da velocidade ao longo do percurso tipo oito.*

Na Figura 5.75 e na Figura 5.76 estão representadas as evoluções do erro da posição e da orientação, respetivamente, ao longo do tempo para o mesmo percurso, quando comparado com o GPS. Na Figura 5.77 estão representadas as evoluções do erro da velocidade linear, quando comparado com a odometria.

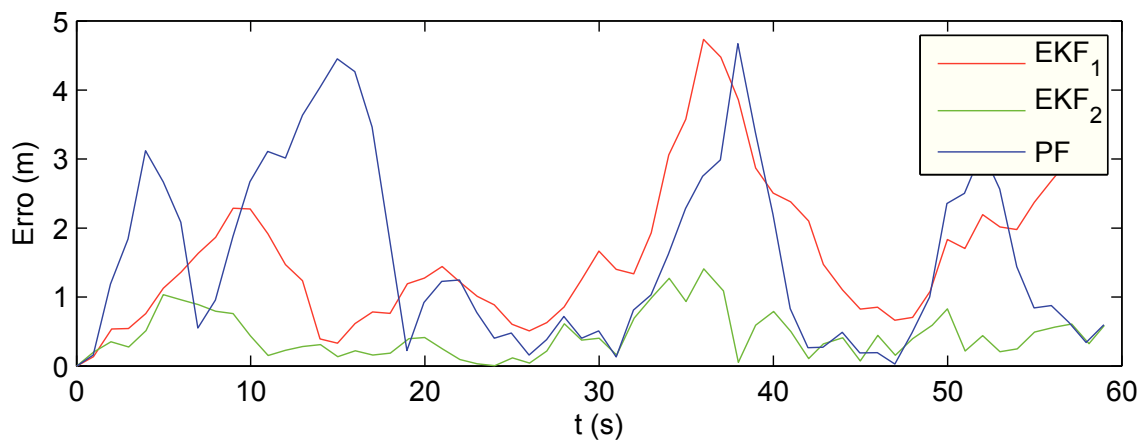


Figura 5.75: Erro de distância ao longo do percurso tipo oito.

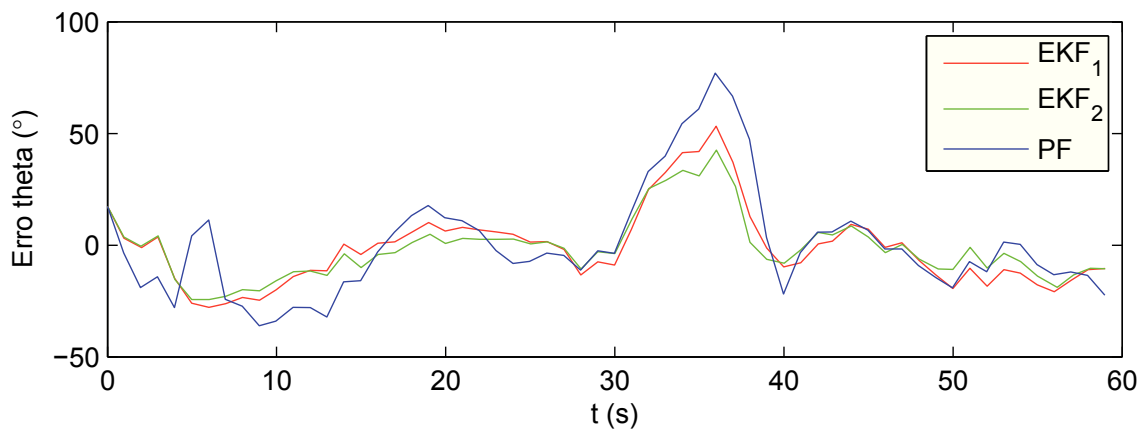


Figura 5.76: Erro da orientação ao longo do percurso tipo oito.

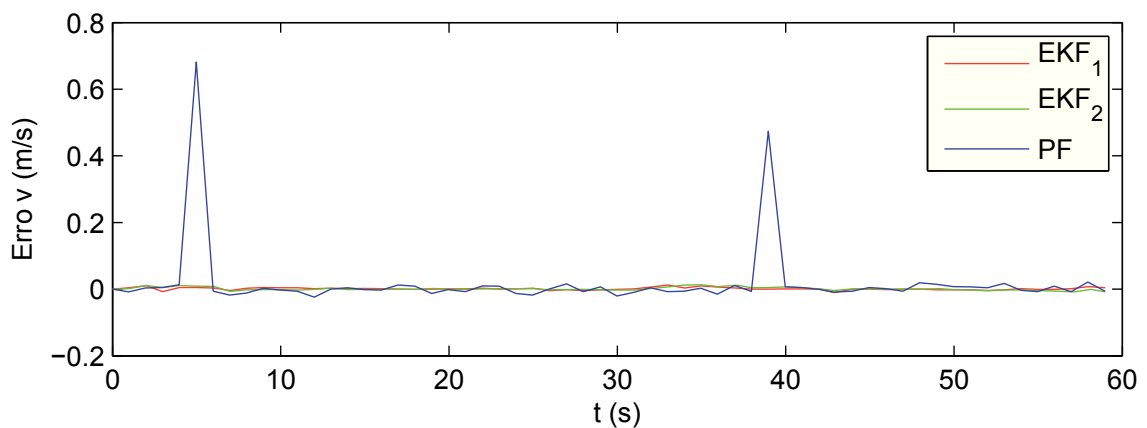


Figura 5.77: Erro da velocidade ao longo do percurso tipo oito.

Os resultados obtidos foram relativamente semelhantes aos do teste do tipo oito, em ambiente real, utilizando a IMU, à exceção do erro da posição que, no método PF e o EKF\_1 obtiveram um erro bastante superior. Na Tabela 5.8 é feita a comparação dos

métodos em estudo em termos de posição, orientação e velocidade linear, para o percurso do tipo oito.

### Comparação para o percurso tipo oito em ambiente real:

Na Tabela 5.8 está representado o desvio padrão e a média do erro, obtidos para os métodos em estudo, no percurso do tipo oito em ambiente real, fazendo a distinção entre a utilização e a não utilização da IMU.

Tabela 5.8: Comparação entre algoritmos na simulação para o percurso do tipo oito.

	Método	Utilizando IMU		Não utilizando IMU		Un
		Desvio Padrão	Erro médio	Desvio Padrão	Erro médio	
Posição	EKF_1	1.11328	1.84476	1.09139	1.62667	$m$
	EKF_2	0.80510	0.67026	0.051152	0.44602	
	PF	0.99008	0.85603	1.55473	1.61683	
Orientação	EKF_1	19.47906	-4.65274	17.40014	-1.22358	° (grau)
	EKF_2	15.57159	-3.66029	14.45199	-1.44604	
	PF	11.06899	-0.51654	25.75119	0.28314	
Velocidade linear	EKF_1	0.01635	0.02916	0.00479	0.00119	$m/s$
	EKF_2	0.00684	0.00111	0.00685	0.00098	
	PF	0.29523	0.02716	0.15103	0.01887	

Mais uma vez, a vermelho está representado o método que apresentou melhores resultados em termos de posição, a verde em termos de orientação e a azul de velocidade linear.

Continuando com o EKF\_2 sem utilização da IMU como o melhor resultado em termos de posição, neste teste este método também mostrou a sua eficiência em velocidade, obtendo no menor erro médio neste teste. Na orientação existem algumas divergências nos erros resultantes, sendo o método PF sem utilização da IMU a melhor opção.

### Teste de utilização normal

Num comprimento de aproximadamente 110 m, este teste foi realizado num ambiente típico citadino, tendo tido várias paragens, acelerações mais bruscas e manobras, sendo assim um teste mais orientado para a utilização final.

Este percurso não foi efetuado no mesmo local que os testes anteriores, estando representado na Figura 5.44 o local onde utilizado para realização do teste.

### - Teste de utilização normal utilizando Odometria, GPS e IMU

A Figura 5.78(a) representa o teste realizado utilizando o EKF\_1, a Figura 5.78(b) com o EKF\_2 e a Figura 5.78(c) o PF. Na Figura 5.78(d) estão sobrepostos os métodos em análise sobre uma fotografia via satélite.

Mais uma vez, a trajetória estimada pelo método em estudo é representada a azul; a trajetória adquirida pelo GPS a vermelho; e a cor amarela representa as várias partículas estimadas, no filtro de partículas. Assim como nos testes anteriores, realizados em ambiente real, na figura de sobreposição dos resultados em mapa (Figura 5.78(d)) o percurso foi deslocado de modo a que o percurso efetuado correspondesse com o demonstrado, caso contrário estaria ligeiramente desviado, sendo que este desvio deve-se principalmente ao erro associado ao GPS (ver Secção A.2.2), isto porque não foi utilizado um GPS de precisão nem qualquer estação de base.

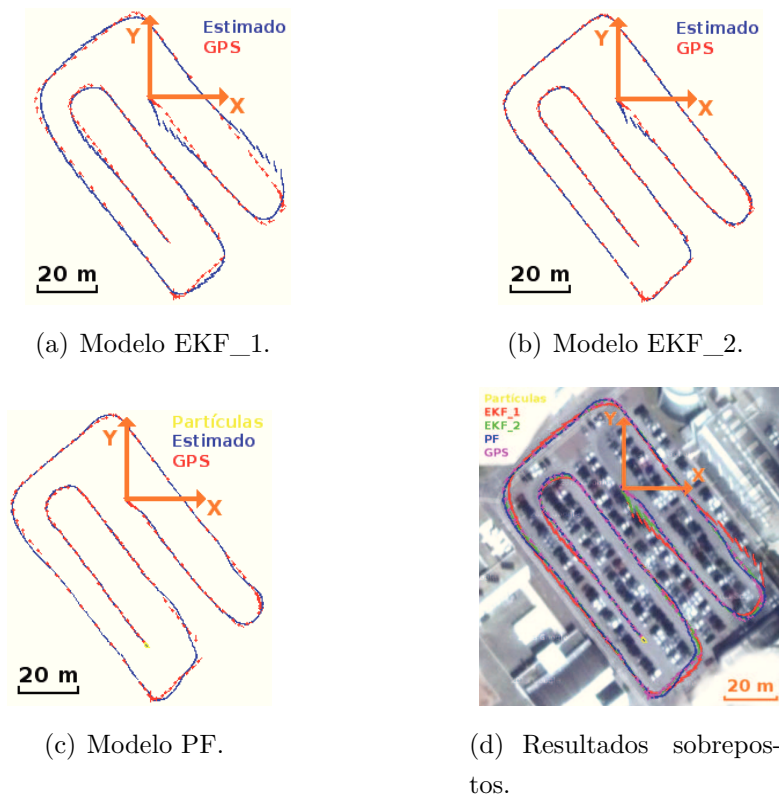


Figura 5.78: *Teste em ambiente real com utilização normal.*

Para melhor comparação dos métodos são representados em gráfico os parâmetros do estado. Na Figura 5.79, Figura 5.80, Figura 5.81 e na Figura 5.82, referente a  $x_{k+1}$ ,  $y_{k+1}$ ,  $\theta_{k+1}$  e  $v_{k+1}$ , respetivamente, de cada método.

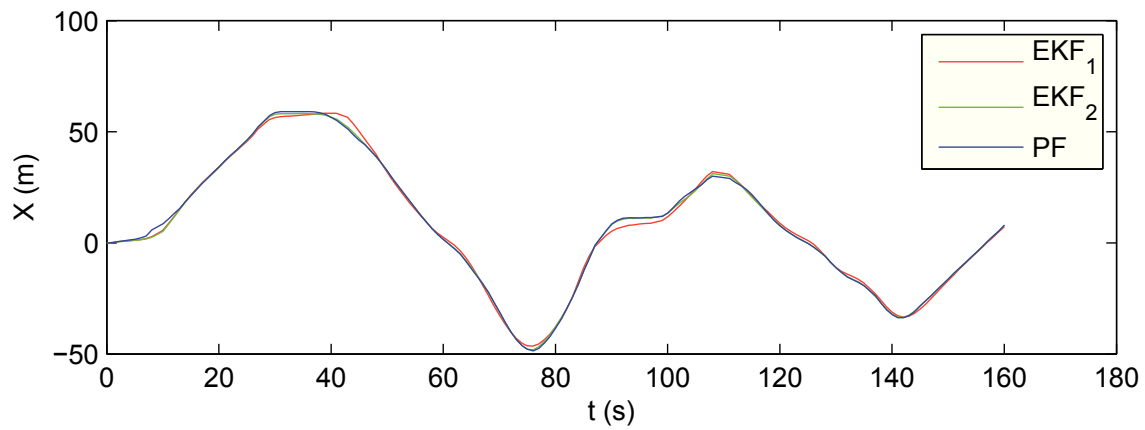


Figura 5.79: *Evolução da posição em X ao longo do percurso de utilização normal.*

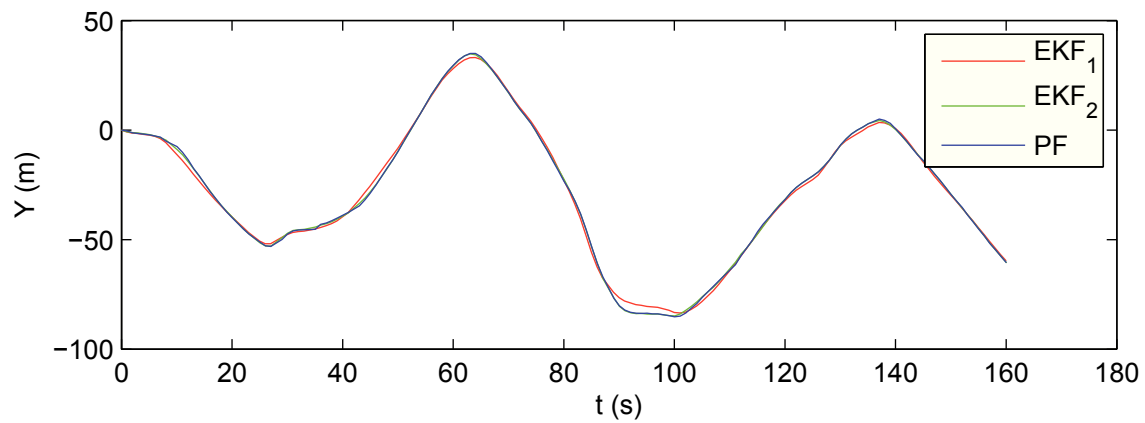


Figura 5.80: *Evolução da posição em Y ao longo do percurso de utilização normal.*

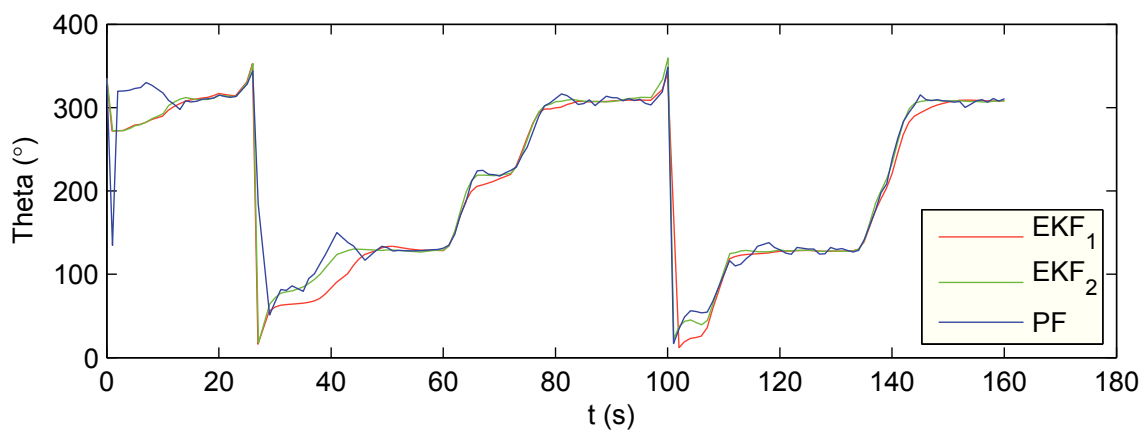


Figura 5.81: *Evolução da orientação ao longo do percurso de utilização normal.*

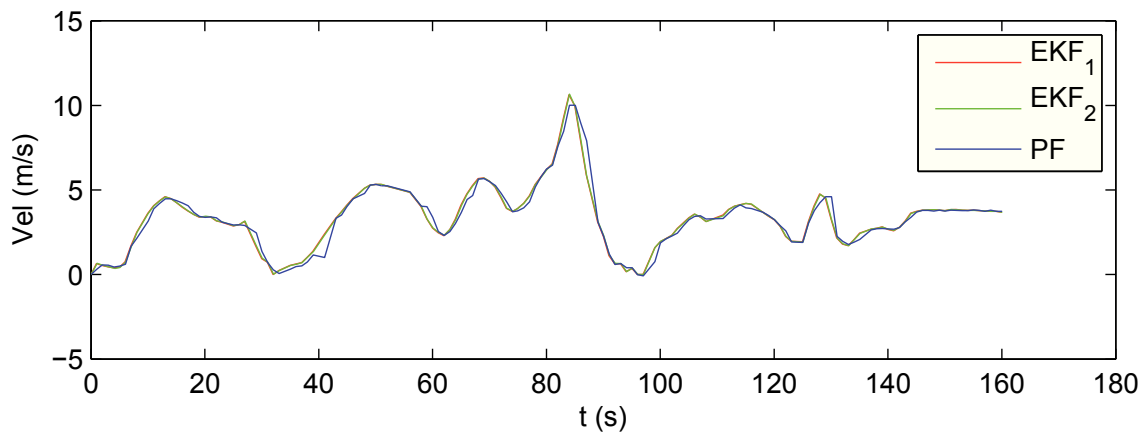


Figura 5.82: *Evolução da velocidade ao longo do percurso de utilização normal.*

Tendo em conta os resultados obtidos nos testes anteriores, as grandes variações, em velocidade linear e em orientação do veículo, ocorridas durante o percurso, faziam esperar maior distinção de resultados entre os métodos, o que, visualmente, não se verificou. Para melhor comparação, na Figura 5.83 e na Figura 5.84 estão representadas as evoluções do erro de posição e orientação ao longo do tempo para o mesmo percurso, quando comparado com o GPS, e na Figura 5.85 as evoluções do erro de velocidade linear ao longo do tempo, quando comparado com a odometria.

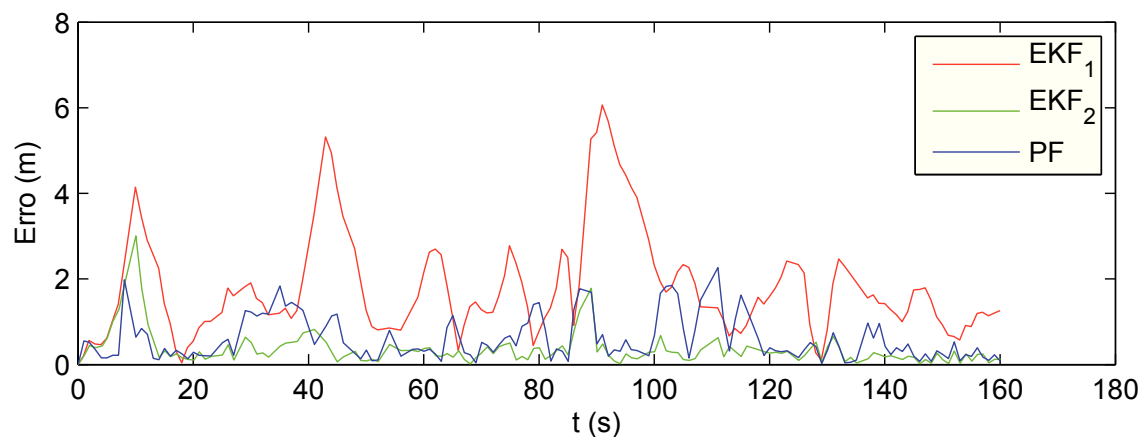


Figura 5.83: *Erro de distância ao longo do percurso de utilização normal.*

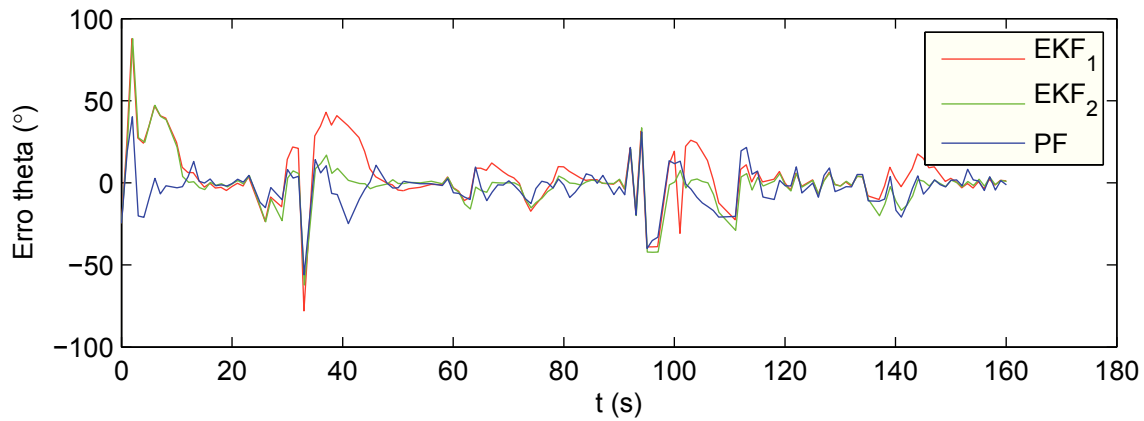


Figura 5.84: Erro da orientação ao longo do percurso de utilização normal.

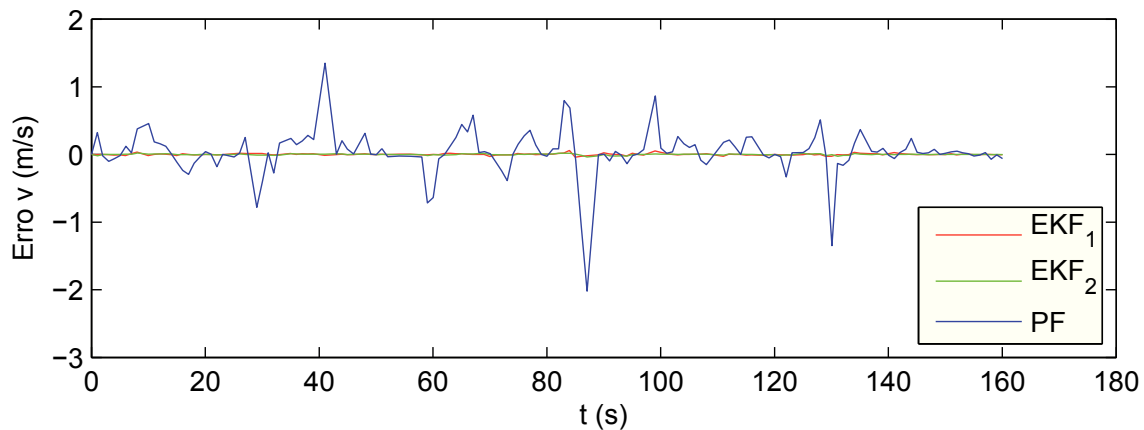


Figura 5.85: Erro da velocidade ao longo do percurso de utilização normal.

Mais uma vez, o EKF\_1 mostrou ter resultados bastante irregulares e com erros elevados em posição, chegando mesmo a ultrapassar os 6 m de erro. Ainda sobre o erro em posição, o EKF\_2 e o PF demonstraram ter resultados aproximados e aceitáveis, para a utilização pretendida. No erro em orientação, os resultados obtidos foram relativamente semelhantes entre métodos, tendo ambos obtido variações de erro relativamente altas. Através da Figura 5.81, é possível observar que o erro da orientação é influenciado pelas variações da orientação do veículo. Como era de esperar, devido à importância associada à velocidade linear adquirida pela odometria, o erro médio em velocidade é quase nulo, quando utilizada a odometria como referência. Isto não se verifica no método PF pois neste método é atribuída mais importância à posição.

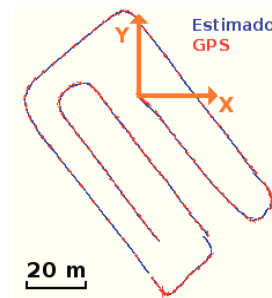
Na Tabela 5.9 é feita a comparação dos métodos em estudo em termos de posição, orientação e velocidade linear, para o percurso de utilização normal em ambiente real.

- Teste de utilização normal utilizando Odometria e GPS, não utilizando a IMU

A Figura 5.86(a) representa o teste realizado utilizando o EKF\_1, a Figura 5.86(b) com o EKF\_2, a Figura 5.86(c) o PF e a Figura 5.86(d) ambos os modelos sobrepostos em mapa.



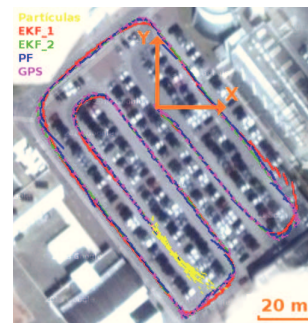
(a) Modelo EKF\_1.



(b) Modelo EKF\_2.



(c) Modelo PF.



(d) Resultados sobrepostos.

Figura 5.86: Teste em ambiente real com utilização normal.

De novo, para melhor comparação dos métodos são representados em gráfico os parâmetros do estado. Na Figura 5.87, Figura 5.88, na Figura 5.89 e na Figura 5.90, referente a  $x_{k+1}$ ,  $y_{k+1}$ ,  $\theta_{k+1}$  e  $v_{k+1}$ , respetivamente, de cada método.

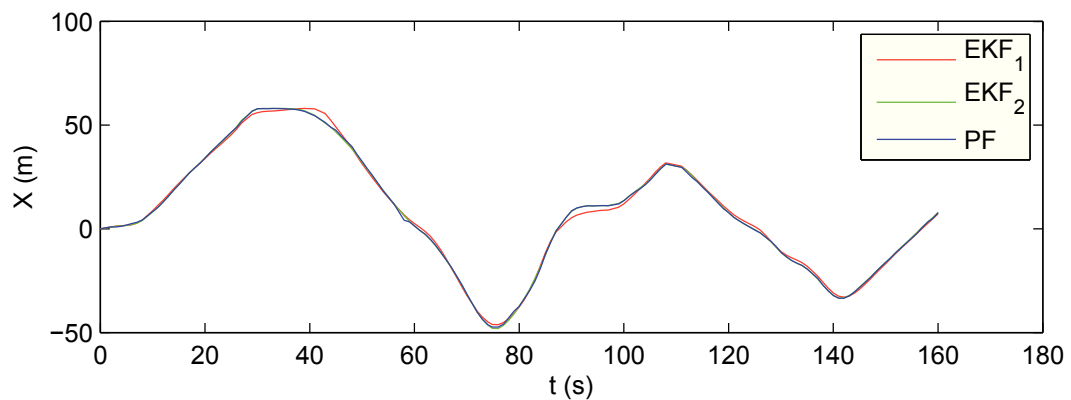


Figura 5.87: Evolução da posição em X ao longo de utilização normal.

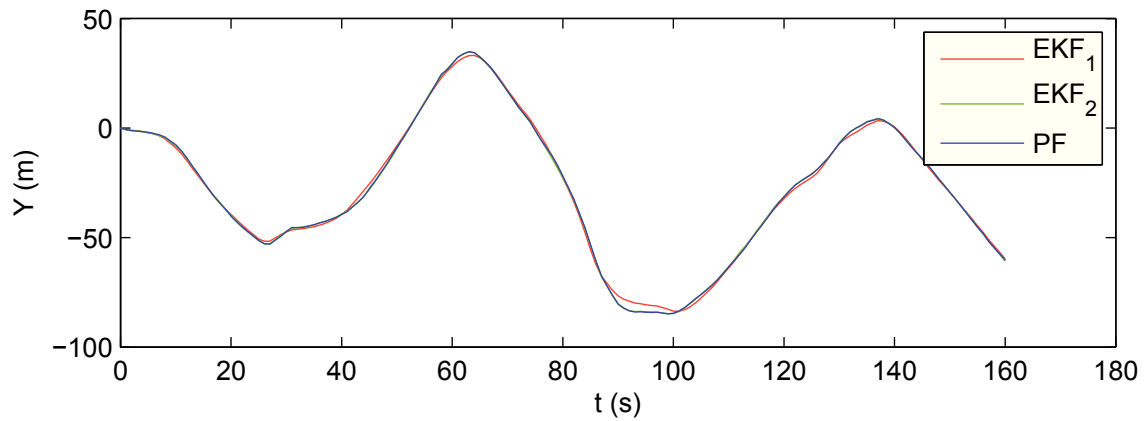


Figura 5.88: *Evolução da posição em Y ao longo de utilização normal.*

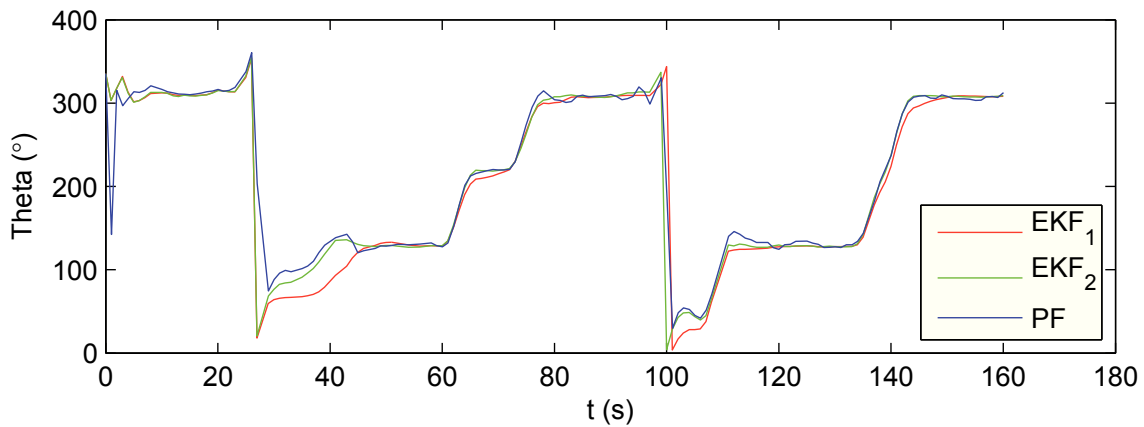


Figura 5.89: *Evolução da orientação ao longo de utilização normal.*

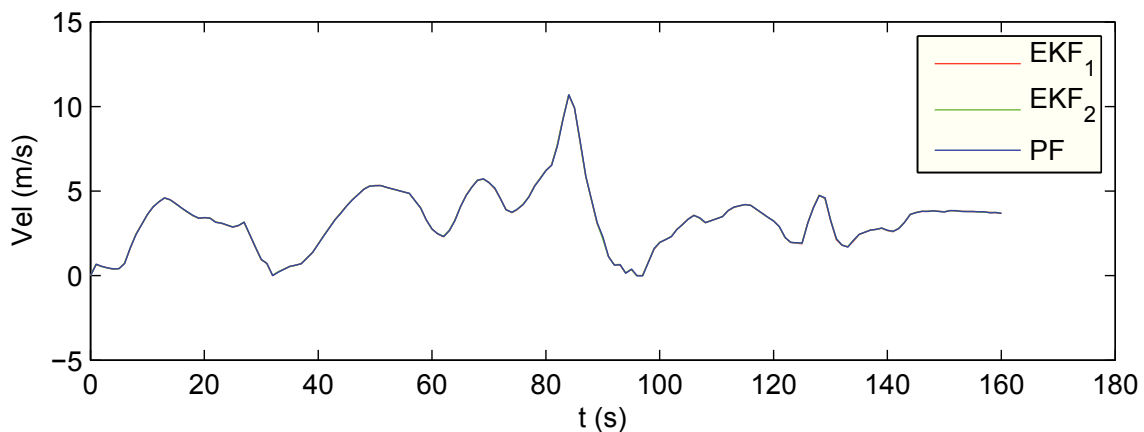


Figura 5.90: *Evolução da velocidade ao longo de utilização normal.*

Bastante semelhantes ao teste anterior, os resultados destes testes obtiveram maior semelhança entre métodos. Para melhor interpretar a diferença entre eles, na Figura 5.91 e na Figura 5.92 estão representadas as evoluções do erro da posição e orientação ao longo do tempo para o mesmo percurso, quando comparado com o GPS e, na Figura 5.93,

as evoluções do erro da velocidade linear ao longo do tempo, quando comparado com a odometria.

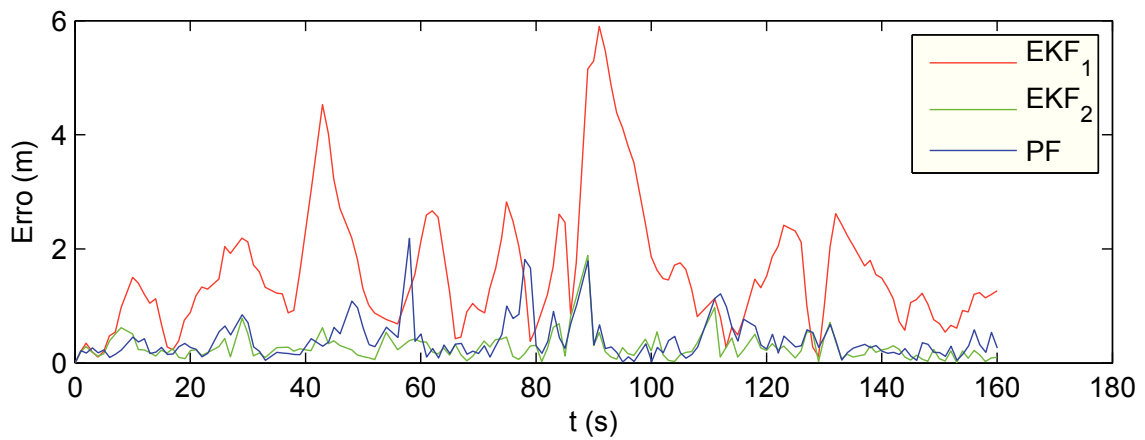


Figura 5.91: Erro de distância ao longo do percurso de utilização normal.

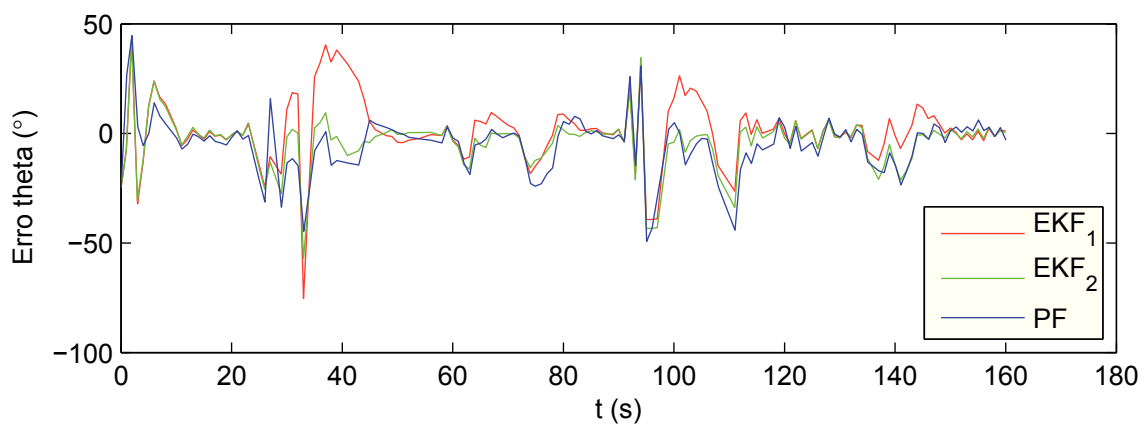


Figura 5.92: Erro da orientação ao longo do percurso de utilização normal.

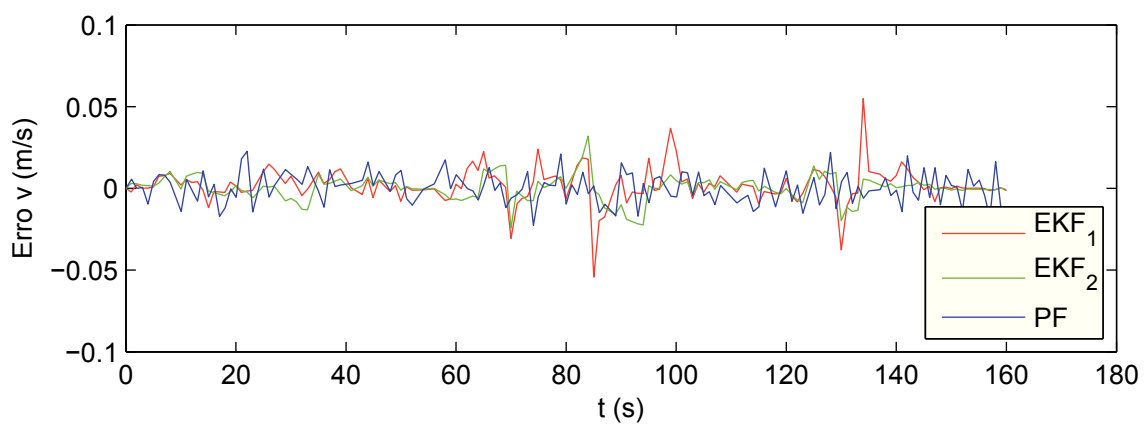


Figura 5.93: Erro da velocidade ao longo do percurso de utilização normal.

Houve claramente uma melhoria no PF em todos os parâmetros. As melhorias obtidas, ao não utilizar a IMU, foram principalmente no método PF, nomeadamente na posição

e na velocidade linear. Em relação ao erro da orientação, pode-se considerar que não houve alteração. Na Tabela 5.9 é feita a comparação dos métodos em estudo em termos de posição, orientação e velocidade linear, para o percurso do tipo oito.

### Comparação para o percurso de utilização normal em ambiente real:

Na Tabela 5.9 está representado o desvio padrão e a média do erro, obtidos para os métodos em estudo, no percurso de utilização normal, fazendo a distinção entre a utilização e a não utilização da IMU.

Tabela 5.9: Comparação entre algoritmos na simulação para o percurso de utilização normal.

	Método	Utilizando IMU		Não utilizando IMU		Un
		Desvio Padrão	Erro médio	Desvio Padrão	Erro médio	
Posição	EKF_1	1.23836	1.78428	1.14412	1.46703	m
	EKF_2	0.48658	0.35403	0.39117	0.27727	
	PF	0.57248	0.60251	0.50299	0.46728	
Orientação	EKF_1	18.04293	4.71240	15.44316	1.67502	° (grau)
	EKF_2	15.70773	-0.42977	12.53710	-3.64361	
	PF	12.00782	-2.85733	13.12035	-5.48317	
Velocidade linear	EKF_1	0.01479	0.00195	0.01313	0.00246	m/s
	EKF_2	0.01270	-0.00002	0.01116	0.00009	
	PF	0.39709	0.02685	0.01325	0.00025	

De novo, a vermelho está representado o método que apresentou melhores resultados em termos de posição, a verde em termos de orientação e a azul de velocidade linear.

Num percurso citadino, o EKF\_2 demonstrou claramente ser o método com menores erros médios em quase todos os parâmetros, tendo sido, de novo, o EKF\_2 sem utilização da IMU a melhor escolha em posição, e o EKF\_2 com utilização da IMU o melhor resultado em orientação e em velocidade. Dando preferência à posição, pode-se assumir que o método EKF\_2 sem utilização da IMU, neste teste, seria a melhor escolha.

Na Tabela 5.10 é apresentada a apreciação final dos métodos em análise, para os testes em ambiente real. Para isso, é apresentado a média e o desvio padrão para os testes realizados em ambiente real.

Não esquecendo que todos os valores utilizados são com referência à odometria e ao GPS, deixando por isso a IMU em desvantagem, podemos estar assim a comparar com valores mais errados que o filtro. Assumindo as referências sem erro, o EKF\_2 sem utilização da IMU demonstrou, ao longo dos testes efetuados, ser o método com menor

Tabela 5.10: *Comparação entre algoritmos para todos os testes efetuados em ambiente real.*

	Método	Utilizando IMU		Não utilizando IMU		Un
		Desvio Padrão	Erro médio	Desvio Padrão	Erro médio	
Posição	EKF_1	1.16258	1.72471	1.11697	1.56518	$m$
	EKF_2	0.57123	0.44270	0.43121	0.34577	
	PF	0.78654	0.72642	1.09327	0.85370	
Orientação	EKF_1	18.60607	0.49075	16.45667	-0.41786	° (grau)
	EKF_2	15.41279	-2.33237	13.51763	-3.27514	
	PF	12.00156	-2.29951	17.45867	-3.81294	
Velocidade linear	EKF_1	0.01832	0.00826	0.01037	0.00180	$m/s$
	EKF_2	0.01100	0.00035	0.01028	0.00061	
	PF	0.33163	0.02026	0.07500	0.00424	

erro médio associado, o que é aqui de novo confirmado. É também possível verificar que o método EKF\_1 sem utilização da IMU foi o filtro que obteve menor erro médio, nos testes realizados em ambiente real. Na velocidade linear, o EKF\_2 com utilização da IMU foi o método que se distinguiu pelo menor erro em média. Como referido, neste trabalho é dada principal importância à posição, seguindo-se da orientação, e por último à velocidade linear, pelo que, o EKF\_2 com utilização da IMU e o EKF\_2 sem utilização da IMU seriam as melhores escolhas. Para ter uma melhor noção da diferença entre estes dois métodos, na Figura 5.94 é apresentada a sobreposição dos mesmos em mapa. Nesta Figura 5.94 o EKF\_2 com utilização da IMU é representado a azul, o EKF\_2 sem utilização da IMU é representado a verde e o percurso medido pelo GPS (referência) é representada em magenta.

Figura 5.94: *Método EKF\_2 no teste de utilização normal, com e sem IMU.*

Como é possível perceber, ambos os filtros têm comportamentos muito semelhantes, tendo contudo alguns resultados diferentes, nomeadamente no início do percurso e em

algumas curvas. Privilegiando a posição, e assumindo as referências corretas, o método que obteve melhores resultados, tendo em conta os critérios escolhidos, foi o EKF\_2 sem utilização da IMU.

Resumindo, em ambos os ambientes testados, o EKF\_2 sem utilização da IMU foi o método que obteve melhores resultados. A IMU, de um modo geral, aparenta tornar o sistema menos robusto, podendo dever-se ao facto da utilização do GPS e da odometria como referências, tendo por isso tendência a prevalecer estes sensores. Em contra-partida, a IMU é um sensor sensível a diversos fatores externos presentes no quotidiano, podendo também ter sido influenciada por eles. Os fatores externos que influenciam o seu funcionamento podem ser inúmeros, desde a simples trepidação do veículo, provocada pela vibração do motor e pela sinuosidade da estrada, à presença de materiais geradores de campos magnéticos, por exemplo materiais metálicos, dispositivos eletrónicos, condutores elétricos ou cabos de alta tensão. Assim, não estando disponível um dispositivo de medida real neste ambiente, assumindo que as referências (GPS e odometria) estão corretas, podemos assumir que, para um modelo 2D em ambiente real, de um modo geral, a IMU não é uma boa opção.

# Capítulo 6

## Conclusões e Trabalho Futuro

Neste trabalho foram apresentadas duas implementações possíveis para a localização de um veículo automóvel baseadas no método EKF e uma baseada no método PF. Em todos os casos foram utilizados três fontes de informação sensorial: odometria, IMU e GPS. Numa das implementações do EKF a odometria foi utilizada na sua forma clássica, no passo de correção, enquanto na outra implementação foi utilizada como sinal de controlo. Os métodos desenvolvidos foram testados num ambiente de simulação 3D, recorrendo ao Gazebo, e em ambiente real, recorrendo a uma Volkswagen Sharan alterada para o efeito.

Os testes realizados mostraram que os resultados obtidos no ambiente real foram quase tão bons como os obtidos no ambiente de simulação, não esquecendo que, em ambiente real, devido à inexistência de um sistema com referência real, foram utilizados alguns sensores como referência, nomeadamente o GPS para posição e orientação e a odometria para velocidade linear. Os resultados, em ambiente de simulação, foram obtidos melhores resultados devido a, no modelo do veículo virtual utilizado, não terem sido considerado alguns fatores externos que possam prejudicar o desempenho dos sensores, nomeadamente ter diferentes pressões nos pneumáticos do veículo (influencia a velocidade medida e induz deslocamento no ângulo das rodas direcionais dos sensores do veículo), a presença de linhas elétricas e a passagem de veículos nas proximidades (geram campos magnéticos que podem induzir os magnetómetros da IMU em erro), a trepidação do veículo (a trepidação influencia os valores acelerómetros e giroscópios da IMU) e a proximidade de edifícios (prejudica o sinal do GPS, influenciando na sua precisão). Verificou-se também que o ângulo de direção dos sensores do veículo tem um erro associado, que varia com a orientação das rodas, sendo por isso interessante, num trabalho futuro, implementar um sistema para a sua calibração rigorosa, podendo assim levar a melhorar o desempenho do sistema. Os resultados mostraram também que, para os testes efetuados, de um modo geral, todos os filtros tiveram valores aproximados e aceitáveis, tendo-se quase sempre distinguido o EKF\_2 (filtro EKF com odometria no passo de predição) como método de

localização com menor erro de estimação, em média. Em contrapartida, o PF demonstrou ser um filtro com menor tempo de convergência quando o estado estimado tem um erro médio superior, isto é, por exemplo, no início de cada percurso, o PF converge para o estado real num tempo inferior ao das outras implementações aqui analisadas. A IMU, em modelo 2D, demonstrou não ser uma boa opção para a estimativa do estado, tendo obtido em quase todos os testes um maior erro, em média. De salientar que não foi implementado nenhum método para dar o valor real do estado nos testes com o veículo real, ou uma estimativa com um valor com uma precisão elevada/controlada. Ao utilizar o GPS e a odometria como referência, nos testes com o veículo real, sabendo que lhes foi atribuída uma maior confiança, estes podem estar a fazer o estado estimado convergir para um valor que não é real, não obstante a calibração prévia dos sensores. Num projeto futuro, no âmbito da condução autónoma, seria vantajoso ter um sistema de localização real para testes, conseguindo assim obter resultados mais precisos e de maior confiança. Este sistema poderia ser desenvolvido com base em câmaras fixas no exterior, ou através de sensores GPS diferencial com estação base, apenas para efeitos de avaliação e validação dos resultados.

A fim de aperfeiçoar e prosseguir o trabalho desenvolvido, propõe-se no futuro estender o modelo desenvolvido para 3D, permitindo assim a utilização do sistema de localização sem limites. Nessa situação, poderá ser uma melhor opção a utilização de coordenadas geográficas (latitude, longitude e altitude) ao invés das coordenadas cartesianas, aqui utilizadas. Poderá também ser avaliada a possibilidade de estudo de outras abordagens, por exemplo, o método UKF [36, 37, 38] que, segundo os autores de [22], em média tem um menor erro de estimativa, ou ainda a utilização de diferentes filtros para diferentes parâmetros no estado, por exemplo utilizando o EKF para a posição e o PF para a orientação [39], de modo a usufruir das vantagens de cada método. Também seria útil a alteração do sistema MCBN de forma a disponibilizar mais informação, nomeadamente a mudança engrenada. Sabendo a mudança engrenada, acrescentaria à aplicação a possibilidade de funcionar nos dois sentidos (marcha a frente e marcha a trás) em ambiente real. Com o objetivo futuro de navegação, este sistema, como está, não é ainda suficiente para navegação, sendo necessária a inclusão de sensores para deteção de obstáculos e área livre, como laser e/ou câmaras, que mesmo na presença deste erro, permitirão ajustar a estimativa do veículo às zonas de circulação mais próximas e desta forma permitir a navegação.

Implementando as propostas anteriores, a visualização num mapa, em tempo real, seria uma aposta interessante. Esta funcionalidade não traria robustez ao sistema, mas melhoraria a perceção em termos visuais. Consistiria na interligação da aplicação com um mapa virtual, por exemplo o Google Earth ou um mapa de GPS, para poder visualizar em tempo real a posição do veículo.

Em suma, os resultados obtidos consideram-se muito positivos. Ainda assim, prevê-se

uma melhoria dos resultados com uma calibração mais rigorosa dos sensores.



# Bibliografia

- [1] Fink e D. Aleksandrovich, “The Prevention of Road Accidents in the Countries of Europe, Asia and the United States of America,” *World Applied Sciences Journal*, vol. 30, no. 12, 2014.
- [2] E. Ackerman, “Google’s Autonomous Car Takes To The Streets,” *IEEE Spectrum*, 2010.
- [3] T. Cowen, “Can I See Your License, Registration and CPU?” *New York Times. New York Times*, vol. 28, 2011.
- [4] I. A. Budisteanu, “Using Artificial Intelligence to create a low cost self-driving car,” University of Bucharest, Faculty of Psychology and Educational, Relatório Tec., 2013.
- [5] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, e P. Mahoney, “Stanley: The robot that won the DARPA Grand Challenge,” *Journal of field Robotics*, vol. 23, no. 9, 2006.
- [6] M. Buehler, K. Iagnemma, e S. Singh, “The 2005 DARPA Grand Challenge,” *Springer Tracts in Advanced Robotics*, vol. 36, no. 5, 2007.
- [7] Bertozzi, Massimo, Bombini, Luca, Broggi, Alberto, Buzzoni, Michele, Cardarelli, Elena, Cattani, Stefano, Cerri, Pietro, Coati, Alessandro, Debattisti, Stefano, A. Falzoni *et al.*, “VIAC: An out of ordinary experiment,” in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, 2011.
- [8] A. Broggi, M. Bertozzi, A. Fascioli, e G. Conte, *The Experience of the ARGO Autonomous Vehicle*. World Scientific, 1999.
- [9] D. Braid, A. Broggi, e G. Schmiedel, “The TerraMax autonomous vehicle,” *Journal of Field Robotics*, vol. 23, no. 9, 2006.

- 
- [10] P. Grisleri e I. Fedriga, “The BRAiVE Autonomous Ground Vehicle Platform,” in *IFAC Symposium on intelligent autonomous vehicles*, vol. 7, no. 1, 2010.
- [11] Q. Chen, U. Ozguner, e K. Redmill, “Ohio State University at the 2004 DARPA Grand Challenge: developing a completely autonomous vehicle,” *Intelligent Systems, IEEE*, vol. 19, no. 5, 2004.
- [12] E. P. A. Costa, “Navegação em Ambientes Exteriores do Robô da Série Atlas 2000,” Dissertação de Mestrado, Universidade de Aveiro Departamento de Engenharia Mecânica, 2012.
- [13] P. M. G. Salvado, “Reconstrução Dinâmica de Mapa Local para o ATLASCAR,” Dissertação de Mestrado, Universidade de Aveiro Departamento de Engenharia Mecânica, 2012.
- [14] N. Rodrigues, “Sistema de controlo de baixo nível de um veículo automóvel para executar directivas de condução autónoma,” Dissertação de Mestrado, Escola Superior de Tecnologia e Gestão de Leiria, 2011.
- [15] J. Borenstein, H. R. Everett, e L. Feng, *Navigation mobile robots: System and Techniques*. A. K. Peters, Ltd., 1996.
- [16] J. F. C. Fonseca, “Aplicação de técnicas de fusão/integração sensorial de dados no levantamento do relevo de objectos,” Tese de Doutoramento, Universidade do Minho - Portugal, 1999.
- [17] I. Rekleitis, “Cooperative Localization and Multi-Robot Exploration,” Tese de Doutoramento, School of Computer Science McGill University, Montréal, 2003.
- [18] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Journal of basic Engineering*, 1960.
- [19] G. Welch e G. Bishop, “An Introduction to the Kalman Filter,” University of North Carolina at Chapel Hill, Relatório Tec., 1997.
- [20] P. S. Maybeck, “Stochastic models, estimation, and control, Volume 1,” Institute of Technology Wright-Patterson Air Force Base Ohio, Relatório Tec., 1979.
- [21] J. D. G. Cardoso, “Sistema de localização baseado em DGPS para aplicações em veículos autónomos,” Dissertação de Mestrado, Universidade de Coimbra, 2009.
- [22] S. Thrun, W. Burgard, e D. Fox, “Probabilistic Robotics,” *The MIT Press*, 2005.

- [23] J. Georgy, T. Karamat, U. Iqbal, e A. Noureldin, “Enhanced MEMS-IMU/odometer/GPS integration using mixture particle filter,” *GPS Solutions*, vol. 15, 2011.
- [24] M. S. da Costa Couto, “Localização e Navegação entre Robôs Móveis,” Dissertação de Mestrado, Universidade Técnica de Lisboa, 2010.
- [25] S. Cameron e P. Probert, *Advanced Guided Vehicles, Aspects of the Oxford AGV Project*. World Scientific, Singapore, 1994.
- [26] G. Dudek e M. Jenkin, *Computational principles of mobile robotics*. Cambridge University Press, 2000.
- [27] S. Han e J. Wang, “A novel method to integrate IMU and magnetometers in attitude and heading reference systems,” *Journal of Navigation*, vol. 64, no. 4, 2011.
- [28] O. Huisman e R. A. de By, *Principles of Geographic Information Systems: an introductory textbook*, 4th ed. ITC Educational Textbook Series, 2009.
- [29] M. Ligas e P. Banasik, “Conversion between Cartesian and geodetic coordinates on a rotational ellipsoid by solving a system of nonlinear equations,” *Geodesy and Cartography*, vol. 60, 2011.
- [30] *Department of Defense World Geodetic System 1984, Its Definition and Relationships with Local Geodesic Systems*, 3rd ed. National Imagery and Mapping Agency (NIMA), 2000.
- [31] A. Kelly, “Essential Kinematics for Autonomous Vehicles,” The Robotics Institute Carnegie Mellon University, Relatório Tec., 1994.
- [32] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, e A. NG, “ROS: an open-source Robot Operating System,” *ICRA workshop on open source software*, 2009.
- [33] N. Koenig e A. Howard, “Design and Use Paradigms for Gazebo, An Open-Source Multi-Robot Simulator,” *Intelligent Robots and Systems*, 2004.
- [34] V. Kumar, K. P. S. Rana, e A. K. Sinha, *Design, Performance, and Stability Analysis of a Formula-Based Fuzzy PI Controller*, 7th ed. International Journal of Innovative Computing, Information and Control (IJICIC), 2011.
- [35] M. Nery, “Trekking Brasil,” acessado em 26-03-2014. [Online]. Disponível em: <http://trekkingbrasil.com/orientacao-com-bussola-e-mapa-parte-1/>

- 
- [36] F. Gustafsson e G. Hendeby, "Some Relation Between Extended and Unscented Kalman Filters," *IEEE Transactions on Signal Processing*, 2012.
- [37] S. J. Julier e J. K. Uhlmann, "Unscented Filtering and Nonlinear Estimation," *Processing of the IEEE*, 2004.
- [38] G. A. Terejanu, "Unscented Kalman Filter Tutorial," *Department of Computer Science and Engineering*, 2004.
- [39] Seong-hoon e P. Won, "A Kalman/Particle Filter-Based Position and Orientation Estimation Method Using a Position Sensor/Inercial Measurement Unit Hybrid System," *IEEE Transactions on Industrial Electronics*, 2010.
- [40] C. Powers, "CAN Specification v2.0 - Part B," *Motorola MCTG Multiplex Application*, 1995.
- [41] H. L. Harter, *Circular Error Probabilities*, 55th ed. Journal of the American Statistical Association, 1960.
- [42] R. B. Langley, "Dilution of Precision," *University of New Brunswick*, 1999.
- [43] M. Ferguson, "roserial," acessado em 26-03-2014. [Online]. Disponível em: <http://wiki.ros.org/roserial>

# Apêndice A

## Caraterização dos sensores

Neste capítulo descreve-se as principais especificações dos sensores, métodos de calibração, bem como os esquemas de comunicação existentes.

### A.1 Sensores existentes

Fruto do projeto anterior [14], o veículo já está equipado com codificadores nas rodas, originalmente pertencentes ao sistema de ABS, e codificadores na coluna de direção. Transformados em velocidade linear e orientação das rodas no MCBN, são posteriormente utilizados para calcular a odometria. Os sensores existentes comunicam utilizando o protocolo CAN. De seguida descrevem-se os detalhes desta comunicação, bem como procedimentos e resultados da caraterização do sensores e caracterização do erro a eles associado.

#### A.1.1 Comunicação

A comunicação entre o veículo e o computador é baseada num barramento CAN, convertida através de um conversor CAN-USB (Figura A.1) para comunicação série. Na Tabela A.1 estão demonstradas as principais caraterísticas de comunicação utilizadas.

Tabela A.1: *Especificações de comunicação dos sensores do veículo.*

Caraterística	Valor
<i>Baud rate</i>	250 kbps
Interface	CAN

A comunicação foi configurada no formato *standard data frame* [40], o qual utiliza 11 *bits* para a identificação das tramas (SID). Entre as cadeias de caracteres disponibilizadas



Figura A.1: *Ligação do MCBN ao PC.*

pelo MCBN, a única utilizada foi a presente na Figura A.2, que corresponde à velocidade linear e ao ângulo das rodas. De referir que estas cadeias de caracteres vêm ordenadas de acordo com o formato *little-endian*.

SID	Tamanho	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
101	8	Velocidade do veículo (m/s)				Ângulo de direção (°)		

Figura A.2: *Cadeia de caracteres correspondente à velocidade linear e ângulo direção.*

### A.1.2 Caraterização do erro

Nos métodos EKF em análise, foi sempre assumido que o ruído dos sensores é Gaussiano com média nula. Como na prática isso nem sempre acontece, é necessário calibrar o sensor, por forma a que o erro médio do sensor seja aproximadamente zero. No caso dos sensores do veículo, o cálculo da variância do erro dos codificadores das rodas é realizado em separado do codificador do ângulo de direção. A variância dos codificadores das rodas, em velocidade linear, foi calculada com recurso a um banco de potência de referência MAHA LPS 2000. Esta máquina de rolos tem uma precisão de  $\pm 10\%$  que, mesmo sendo um erro elevado, revelou-se ser suficiente para um bom desempenho do sistema desenvolvido. Na Figura A.3 é possível ver o decorrer da calibração do veículo utilizado, Volkswagen Sharan vr6.



Figura A.3: *Teste codificadores das rodas.*

O teste de velocidade linear, consistiu na limitação da velocidade, por parte do banco de ensaio, obrigando assim a que a velocidade do veículo fosse constante. Realizado para diferentes velocidades, e durante um determinado período de tempo, foi possível retirar a média e a variância do erro, isto com o intuito de obter a relação da variância do erro com a velocidade linear. Este teste tem algum erro associado, sendo que a estabilização da velocidade, limitada pelo banco de potência, varia em  $\pm 0.2$  m/s. Os resultados obtidos no teste podem ser visualizados na Tabela A.2. Não se tendo verificado qualquer relação entre variância do erro e a velocidade linear, foi utilizada a variância total do teste.

Como é possível verificar, o sensor não tem erro médio nulo, mas sim um erro médio em função da velocidade. Tendo em conta que, de momento, as velocidades de utilização pretendidas são baixas (em torno dos 20 km/h), foi assumido que o erro médio é nulo. Posteriormente, em trabalhos futuros em que as velocidades sejam superiores, poderá ser necessário implementar um modelo do erro de velocidade do veículo.

Tabela A.2: *Variância e média do erro da velocidade linear.*

	Velocidade linear						
Erro	2.778	5.556	11.111	22.222	33.333	Total	Un
Médio	-0.05180	-0.00831	0.08801	0.16334	0.38073	0.11127	m/s
Variância	0.00086	0.00024	0.00050	0.00049	0.00358	0.01987	m <sup>2</sup> /s <sup>2</sup>

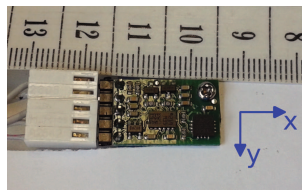
Durante os testes efetuados, verificou-se que o ângulo das rodas tem um deslocamento associado. Assumindo o fator de escala entre a orientação das rodas e a orientação do volante correto (calculado no projeto anterior), assume-se também que a média do erro é constante, e por isso igual para qualquer orientação. Assim, o teste foi realizado recorrendo ao banco de potência. Este obriga a que a direção esteja direita (ângulo das rodas a  $0^\circ$ ), podendo assim retirar a variância e a média/deslocamento do erro. Os resultados obtidos no teste podem ser visualizados na Tabela A.3.

Tabela A.3: *Variância e média do erro do ângulo das rodas.*

Erro	Total	Un
Médio	0.354167	°
Variância	0.0143	(°) <sup>2</sup>

## A.2 Sensores adicionados

Alguns dos sensores pretendidos não estavam ainda integrados no veículo, tendo sido por isso adicionados ao mesmo. Um dos sensores adicionados foi a IMU, com a referência MinIMU-9 v2 (ver Figura A.4). Esta é constituída por três acelerómetros, três giroscópios e três magnetómetros. As suas principais especificações estão na Tabela A.4. Mais à frente encontra-se a descrição do modo comunicação, bem como a sua calibração e a caracterização do erro.

Figura A.4: *IMU - ref: MinIMU-9 v2.*Tabela A.4: *Especificações dos sensores inerciais - ref: MinIMU-9 v2.*

Caraterística	Valor
Tensão funcionamento	2.5 - 5.5 V
Corrente funcionamento	10mA
Interface	I <sup>2</sup> C
Frequência amostragem	100 Hz
Gama de medição	±250, ±500, ou ±2000 °/s (giroscópio) ±2, ±4, ±8, ou ±16 g (acelerómetro) ±1.3, ±1.9, ±2.5, ±4.0, ±4.7, ±5.6, ou ±8.1 gauss (magnetómetro)

O outro sensor adicionado foi o GPS, com referência GPSlim236 (ver Figura A.5). As suas principais especificações são apresentadas na Tabela A.5, enquanto a caracterização do erro associado ao sensor se descreve na secção seguinte.



Figura A.5: GPS - ref: Holux GPSlim236.

Tabela A.5: Especificações do GPS - ref: Holux GPSlim236.

Caraterística	Valor
Tensão funcionamento	4.5 - 5.5 V
Corrente funcionamento	75-85mA
Número máximo de satélites	20
Interface	Série
Frequência amostragem	1 Hz
Formato protocolo	NMEA
Datum	WGS84
Precisão posição	5-25m CEP <sup>1</sup>
Precisão velocidade	0.1 m/s

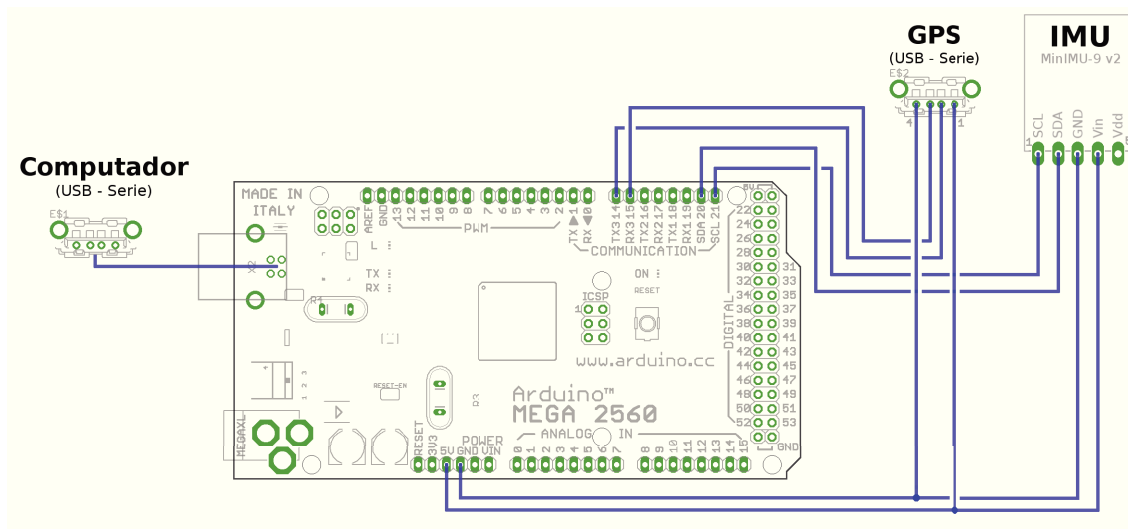
### A.2.1 Comunicação

A comunicação entre a IMU, o GPS e a aplicação foi realizada com recurso a uma placa de desenvolvimento Arduino, mais especificamente o Arduino mega 2560. As principais especificações estão na Tabela A.6. Este dispositivo apenas é utilizado para fazer a aquisição dos dados dos sensores e posteriormente envio para o respetivo nó no ROS. Na Figura A.6 estão representadas as ligações efetuadas para receção da informação sensorial, GPS e IMU, via Arduino.

<sup>1</sup>CEP - Erro circular de probabilidade (*Circular error probable*) [41].

Tabela A.6: *Especificações do Arduino Mega 2560.*

Caraterística	Valor
Tensão funcionamento	4.5 - 5.5 V
Arquitetura	8 bit
Frequência	16 MHz
Flash	256 kBytes
EEPROM	4 kBytes
RAM	8 kBytes
Serial USARTs	4

Figura A.6: *Arduino Mega 2560 e respectivas ligações.*

## A.2.2 Calibração e caracterização do erro

### IMU

A calibração do acelerómetro consistiu na colocação em repouso da IMU com o eixo dos X perpendicular ao solo, em ambas as direções, de modo a calcular o fator de escala correspondente à força da gravidade. Considerando que os acelerómetros não têm deslocamento, o fator de escala médio é 0.00952. O cálculo da variância dos acelerómetros consistiu no mesmo teste que o anterior, mas já utilizando o fator de escala para retirar a variação do erro.

Tabela A.7: *Variância e média do erro dos acelerómetros.*

Erro	Eixo x	Un
Médio	-0.08569	m/s <sup>2</sup>
Variância	0.12572	m <sup>2</sup> /s <sup>4</sup>

A calibração do giroscópio consistiu no cálculo do fator de escala de modo a converter as medidas obtidas em unidades SI. O teste realizado resume-se à imposição de diferentes velocidades angulares constantes, durante um determinado período de tempo, em torno do eixo Z, conseguindo assim extrair a relação entre a velocidade angular real e as medidas obtidas. Neste caso, o valor médio da relação é 112. Este teste foi efetuado utilizando um Pioneer 3-DX, ver Figura A.7. Assumindo a velocidade angular do Pioneer constante e sem erro, já com o fator de escala, foi calculada a variância do erro, resultando em  $\sigma_{giro}^2 = 0.0345 \text{ (rad}^2/\text{s}^2)$ .



Figura A.7: *Calibração dos giroscópios.*

A calibração dos magnetómetros consistiu na definição da sua gama de funcionamento no ambiente a ser utilizado. O teste realizado resume-se à rotação da IMU nos três eixos, de modo a obter os valores mínimos e máximos relativos a cada eixo. Na Figura A.8 é apresentada a nuvem de pontos obtida através dos valores medidos pelos magnetómetros, para uma rotação em torno de Z do sistema de eixos da IMU. O cálculo da variância da bússola é baseado num percurso em linha reta, com orientação constante. Assumindo a orientação constante, foi calculada a variância do erro da orientação  $\sigma_{bussola} = 0.00245$  rad.

## GPS

Dado que não foi implementado nenhum método para verificação da velocidade real, foi considerada a odometria como referência, sabendo de antemão que tem um erro associado. Na Figura A.9 está representada a velocidade obtida pelo GPS e pela odometria, para o mesmo percurso. A Tabela A.8 apresenta a média e variância do erro da velocidade, obtida neste teste. Considerando a odometria perfeita, foi feita a calibração do GPS. Foi realizado um percurso e comparado os dados obtidos entre os dois - Figura A.9.

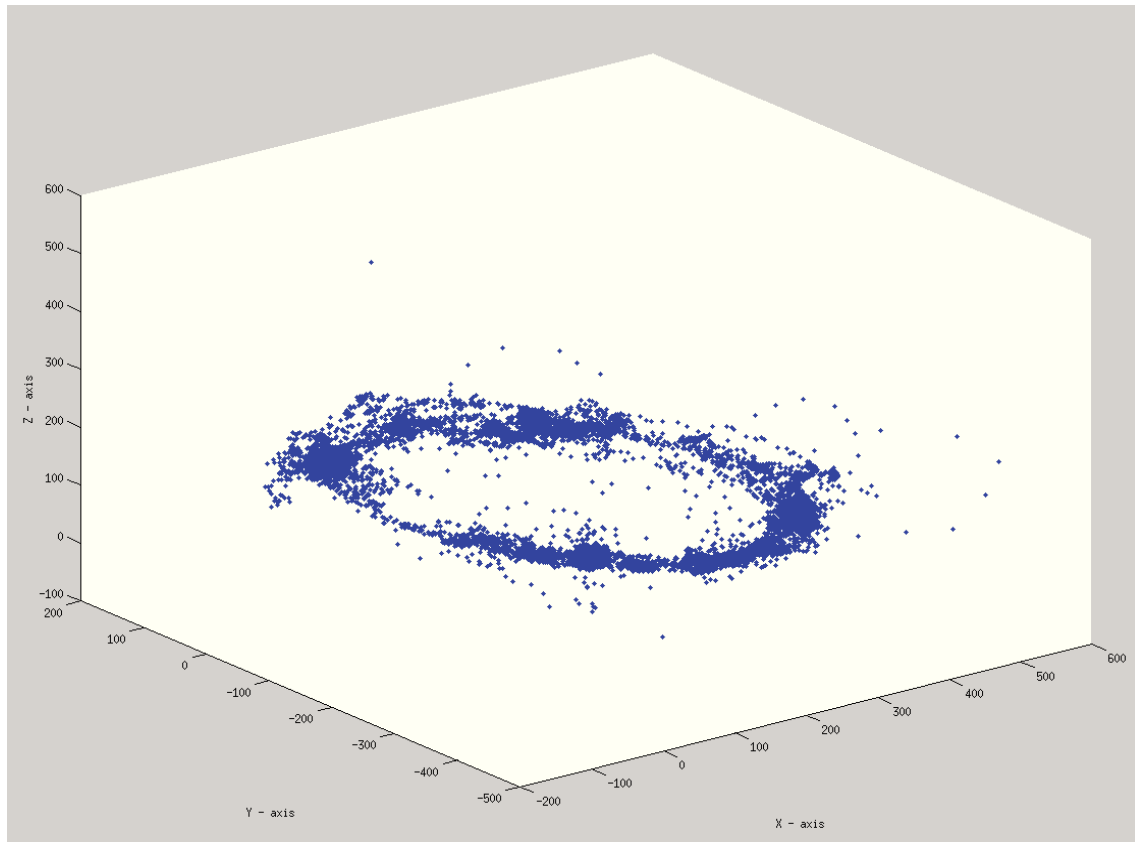


Figura A.8: *Calibração de magnetômetros.*

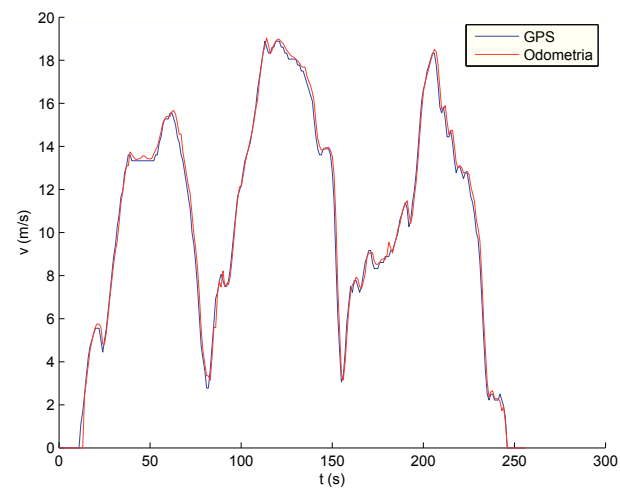


Figura A.9: *Velocidade linear da odometria versus GPS.*

Tabela A.8: *Variância e média do erro da velocidade linear do GPS.*

Erro	Eixo x	Un
Médio	-0.08159	m/s
Variância	0.25689	m <sup>2</sup> /s <sup>2</sup>

O GPS utilizado transmite-nos uma grandeza relacionada com a precisão da localização disponibilizada - HDOP (*Horizontal Dilution Of Precision*) [42], no entanto, após testes efetuados, este valor demonstrou não ser muito exato, decidindo por isso não o considerar. Optou-se então, por calcular a variância do erro em posição, em função de marcos correspondentes a pontos geodésicos conhecidos. Situados na ESTG, estes marcos serviram de referência para calcular o erro em posição, ver Figura A.10. Na Tabela A.9 é apresentado a variância e o erro médio obtido.

Figura A.10: *Pontos geodésicos utilizados para cálculo da variância do erro do GPS.*Tabela A.9: *Variância e média do erro em posição, do GPS.*

Erro	Eixo			Altitude	Un
	X	Y	Z		
Médio	2.07643	0.50756	1.26705	-3.0500015259	m
Variância	4.31154	0.25762	1.60541	8.9559884452	m <sup>2</sup>

### A.2.3 Posicionamento dos sensores adicionados

No modelo definido, considerou-se o sistema de eixos no centro de massa do eixo traseiro do veículo. Sendo o sistema de localização baseado apenas num plano (2D), e por isso não influenciado pela altitude, os sensores adicionados foram posicionados alinhados com o centro do eixo traseiro e no exterior, por forma a melhorar o sinal captado pelo GPS.



Figura A.11: *Posicionamento dos sensores adicionados.*

# Apêndice B

## Manual de utilizador

Este manual de utilizador contém os passos necessários para utilizar o SLVA em ambiente de simulação e em ambiente real. A aplicação SLVA é comum aos dois ambientes, tendo aplicações específicas para a comunicação com o *hardware* (ambiente real e ambiente simulado). As aplicações utilizadas em ambiente de simulação consistem na simulação em Gazebo e no controlo do veículo via teclado (ver Figura 4.3). Em ambiente real foi utilizada a aplicação do Arduino e a aplicação para comunicação com o veículo (ver Figura 4.9).

### B.1 SLVA

O SLVA foi concebido de modo a ser intuitivo na escolha dos sensores e do modelo pretendido. No cabeçalho do "EKF\_Particle\_Filter.hpp" é apresentada a escolha do método (EKF ou PF), do ambiente (simulação ou real), dos sensores (GPS, odometria e IMU) e da consideração da odometria (sensor ou sinal de controlo), ver Figura B.1. É necessário voltar a compilar a aplicação para esta alteração surtir efeito.

```
#define FILTER_IN_USE 1           // (1) EKF, or (0) PF, Current filter in use
#define SIMULATION 0             // (1) Use simulation, or (0) practical essay

#define USE_GPS 1                 // (1) Use, or (0) not, the GPS sensor
#define USE_VEHICLE_SENSORS 1    // (1) Use, or (0) not, the Odometry sensor
#define USE_VEHICLE_SENSORS_PRED 1 // (1) Use, or (0) not, Odometry on prediction step
#define USE_IMU 1                 // (1) Use, or (0) not, the IMU sensor
```

Figura B.1: Configurações do modelo a utilizar no SLVA.



Tabela B.1: *Processo para inicio da aplicação no Gazebo.*

```
# Iniciar Simulacao GAZEBO

# Inicia ROS
# Iniciar apenas se ainda não foi iniciado
roscore&

# Apenas se não estiver definido no bash
# ./usr/share/drcsim-3.0/setup.sh
export GAZEBO_PLUGIN_PATH=$GAZEBO_PLUGIN_PATH:~/<Localizacao>
/MyDRCVehiclePlugin/lib
# Ativa funcionalidades
export VRC_CHEATS_ENABLED=1

# Abre simulação
gazebo -u -s /usr/lib/drcsim-3.0/plugins/libgazebo_ros_api_plugin.so ~/<Localizacao>/MyDRCVehiclePlugin/worlds/my_vehicle.world
```

Tabela B.2: *Processo para inicio da aplicação de atuação.*

```
# Iniciar controlo simulação

# Iniciar apenas se ainda não foi iniciado
#roscore&
# Abre pasta da aplicação
cd <Localização>/Robot_KeyTeleop/bin/
# Iniciação do controlo por teclado
./robot_keyteleop
```

## Arduino

Com o intuito de adquirir a informação sensorial e a retransmitir no respetivo tópico, a aplicação do Arduino pode ser carregada seguindo as seguintes instruções, na respetiva ordem:

1. Abrir programa "GPS\_IMU\_Vehicle" no Arduino;
2. Confirmar escolha do Arduino ("*Tools/Board*") e da porta ("*Tools/Serial Port*");
3. Carregar programa para o Arduino;

Após o programa carregado no Arduino, executar o código da Tabela B.3, num terminal diferente, para correr a implementação de conexão *rosserial* (ROS e porta série) [43].

Tabela B.3: *Processo para início de conexão rosserial.*

```
# Iniciar comunicação rosserial

# Iniciar apenas se ainda não foi iniciado
#roscore&
# Inicia rosserial para a porta ACM0
roscrun rosserial_python serial_node.py /dev/ttyACM0
```

## Comunicação MCBN

Com um objetivo semelhante à aplicação do Arduino, esta também tem como finalidade a aquisição de medidas de sensores, e a sua retransmissão via ROS, no tópico correspondente. Como referido na Secção A.1, os sensores do veículo são adquiridos por comunicação série, sendo retransmitidos via ROS através desta aplicação. Para executar esta aplicação, desde que previamente compilada, usam-se os seguintes passos:

1. Certificar-se que o *roscore* está em execução;
2. Conectar cabo CAN-USB no computador e no MCBN;
3. Correr, num novo terminal, o código presente na Tabela B.4.

Tabela B.4: *Processo para inicio de conexão ao MCBN, sem compilar a aplicação.*

```
# Iniciar comunicação com MCBN

# Iniciar apenas se ainda não foi iniciado
#roscore&
# Abre pasta da aplicação
cd ~/<Localização>/can_serial/bin/
# Iniciação da aplicação para conexão com o veículo
./receiver
```