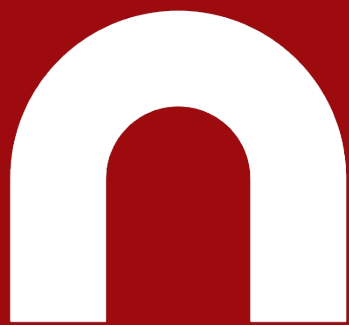




isec
Engenharia

MESTRADO EM ENGENHARIA
ELETROTÉCNICA

**Integração Sensorial para Máquina
Florestal**

Autor

Tiago Manuel Silva Pereira

Orientadores

Professor Doutor Nuno Miguel Fonseca Ferreira

Professor Coordenador com Agregação

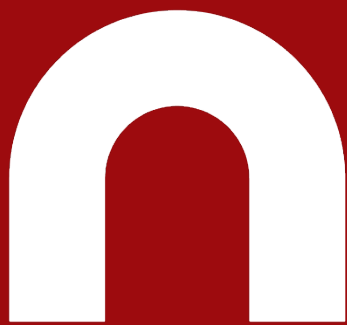
Professor Doutor Victor Daniel Neto dos Santos

Professor Coordenador

Coimbra, outubro de 2023

INSTITUTO POLITÉCNICO DE
COIMBRA

INSTITUTO SUPERIOR
DE ENGENHARIA
DE COIMBRA



isec

Engenharia

DEPARTAMENTO DE ENGENHARIA
ELETROTÉCNICA

Integração Sensorial para Máquina Florestal

Relatório de Trabalho de Projeto para a obtenção do grau de
Mestre em Engenharia Eletrotécnica

Especialização em Automação e Comunicações em Sistemas
Industriais

Autor

Tiago Manuel Silva Pereira

Orientador

Professor Doutor Nuno Miguel Fonseca Ferreira

Professor Coordenador com Agregação

Co-Orientador

Professor Doutor Victor Daniel Neto dos Santos

Professor Coordenador

Supervisor na empresa Universidade de Coimbra (UC)

Professor Doutor Carlos Xavier Pais Viegas

INSTITUTO POLITÉCNICO DE
COIMBRA

INSTITUTO SUPERIOR
DE ENGENHARIA
DE COIMBRA

Coimbra, outubro de 2023

AGRADECIMENTOS

Após um longo percurso enquanto estudante, gostaria de manifestar o meu agradecimento a todos que integraram, de forma direta ou indireta, para o desenvolvimento deste projeto e alcançar os meus objetivos.

Ao meu orientador, Professor Nuno Miguel Fonseca Ferreira, pelo apoio, dedicação, paciência e incentivo prestado ao longo deste trabalho que sem dúvida foi essencial para alcançar os objetivos finais.

Ao meu coorientador, Professor Victor Daniel Neto dos Santos, pelo apoio prestado.

Ao meu supervisor, Professor Carlos Xavier Pais Viegas, por me ter integrado neste projeto de investigação e por toda a ajuda e contribuições prestadas para a construção deste projeto.

Ao Tiago Gameiro pela amizade, apoio prestado ao longo deste trabalho e pelos bons momentos, os quais guardarei na minha memória.

À equipa de laboratório da ADAI, Diogo Valério, Miguel Gaspar, Ricardo Baila, Francesco di Giorgio, Babak Chehreh e Marta Nobre, pelos bons momentos e pelo acompanhamento prestado ao longo do desenvolvimento deste trabalho.

Ao Sr. Engenheiro João Alves, pelo apoio, dedicação e métodos de ensino que contribuíram de forma essencial para alcançar os objetivos deste projeto.

Aos meus pais, Manuel e Isabel, pelo amor, carinho e motivação que me deram e, que sem dúvida, foram essenciais na concretização dos meus objetivos.

Ao resto da minha família por acreditarem em mim e pelo apoio transmitido.

Finalmente, e não menos importante, agradeço à minha namorada, Liliana Costa, pelo amor, paciência, motivação e por sempre estar presente em todos os momentos.

RESUMO

Atualmente, a frequência dos incêndios florestais tem vindo a aumentar gerando assim alterações climáticas que afetam negativamente a qualidade do ar. Assim, uma das formas para mitigar a intensidade de um incêndio florestal passa pela supressão e controlo da vegetação espontânea. Esta remoção pode ser efetuada através de Máquinas Trituradoras Florestais que podem ser controladas à distância, através de um controlo remoto, as quais revelaram ser um ativo valioso. No entanto, trabalhar com este tipo de máquinas é perigoso e requer trabalhadores qualificados, sendo importante a criação de soluções tecnológicas que possam auxiliar os trabalhadores nesta área.

Desta forma, este projeto visa desenvolver um sistema Sensorial, designado por Sentry, que é aplicado neste tipo de máquinas florestais, mais propriamente numa Máquina Trituradora MDB LV600 PRO, com a finalidade de tornar esta máquina autónoma na realização de tarefas de limpeza florestal. O foco principal deste trabalho vai de encontro ao desenvolvimento do Sentry, ou seja, à escolha sensorial e à manipulação dos dados provenientes dos sensores para que possam ser integrados em algoritmos de navegação de trajetórias.

Este trabalho, primeiramente, começou pela pesquisa bibliográfica e documental sobre a robótica móvel, mais especificamente em contexto de agricultura e floresta, seguindo assim a abordagem de tópicos que vão de encontro à aplicação de métodos para viabilizar um conjunto de parâmetros, a fim de se proceder à aplicação em algoritmos de navegação. De seguida, é apresentada a Metodologia proposta, que tende a realizar uma interação entre o ROS e todos os sensores, através da aplicação de filtros e técnicas de fusão sensorial.

Por fim, os resultados e as conclusões permitem averiguar a robustez do sistema quando este é aplicado na máquina trituradora e verificar a precisão de todos os procedimentos propostos.

Palavras-Chave: Robótica Móvel Autónoma, Robótica na floresta, ROS, Fusão Sensorial.

ABSTRACT

Currently, the frequency of forest fires has been increasing, leading to climate changes that adversely affect air quality. Therefore, one way to mitigate the intensity of a forest fire is through vegetation removal. This can be done using Forestry Mulching Machines that can be remotely controlled, which has proven to be a valuable asset. However, working with these machines is dangerous and requires qualified workers, making it important to create technological solutions that can assist workers in this field.

Thus, this project aims to develop a Sensory system, named Sentry, which is applied in these types of forest machines, specifically in a MDB LV600 PRO Mulching Machine, with the purpose of making it autonomous in performing forest cleaning tasks. The focus of this work is on the development of Sentry, i.e., sensor selection and handling of data from the sensors so that they can be integrated into trajectory navigation algorithms.

This work initially started with a bibliographic and documentary research on mobile robotics, specifically in the context of agriculture and forestry, following an approach to topics that support a set of parameters for application in navigation algorithms. Next, the proposed methodology is presented, aiming to facilitate interaction between ROS and all sensors, through the application of filters and sensor fusion techniques.

Finally, the results and conclusions allow us to assess the system's robustness when applied to the mulching machine and verify the accuracy of all proposed procedures.

Keywords: Autonomous Mobile Robotics, Robotics in Forestry, ROS, Sensor Fusion.

ÍNDICE

CAPÍTULO 1: Introdução.....	1
1.1 Motivação.....	2
1.2 Enquadramento.....	3
1.3 Objetivos	4
1.4 Descrição da Máquina Florestal	4
1.5 Estrutura do Projeto	6
CAPÍTULO 2: Estado da Arte.....	7
2.1 Robótica	7
2.1.1 Robôs Móveis	7
2.1.2 Robótica na Agricultura e na Floresta.....	10
2.2 Sistemas de Localização.....	12
2.2.1 Localização Absoluta.....	13
2.2.2 Localização Relativa.....	13
2.3 Representação da Orientação em Robôs Móveis.....	14
2.3.1 Representação da Orientação	14
2.3.2 Representação da Posição.....	16
2.4 Sistemas de Navegação	17
2.4.1 Sistemas de Localização por Satélite	17
2.4.2 Sistemas de Navegação Inerciais.....	19
2.5 Sensores	20
2.5.1 Sensores de Orientação	21
2.5.2 Encoders	23
2.5.3 Sensores Ultrassónicos	24
2.5.4 Sensores LIDAR.....	24
2.5.5 Câmaras	26
2.5.6 GPS	27
2.6 Sistemas de Coordenadas GPS	28
2.6.1 Latitude e Longitude	28
2.6.2 UTM.....	29
2.7 Filtros Digitais.....	29
2.7.1 Vantagens dos Filtros Digitais	30

2.7.2	Classificação de Filtros	30
2.7.3	Tipos de Filtros	34
2.8	Fusão Sensorial	37
2.8.1	Filtro de Kalman.....	38
2.8.2	Filtro Complementar	39
2.9	Visão Computacional	40
2.9.1	Princípio Básico de Funcionamento	41
2.9.2	Principais tarefas na Visão Computacional	41
2.9.3	YOLOv5.....	42
2.10	Sistema Operativo ROS.....	42
CAPÍTULO 3: Metodologia Proposta.....		46
3.1	Estrutura Mecânica	46
3.2	Arquitetura do Hardware	47
3.2.1	Compreensão dos sinais da máquina.....	47
3.2.2	Escolha do Hardware	48
3.3	Arquitetura Sensorial.....	52
3.4	Montagem do Sentry.....	55
3.5	Montagem do Sistema RTK	55
3.6	Integração com o ROS.....	56
3.7	Aquisição de Orientação	56
3.7.1	Calibração do Magnetómetro.....	57
3.7.2	Filtragem de dados	60
3.7.3	Compensação do <i>tilt</i>	63
3.7.4	Cálculo do Heading	65
3.8	Aquisição da Posição.....	65
3.8.1	Filtro SMA.....	66
3.8.1	Filtro de Kalman.....	67
3.9	Algoritmo do Sistema de Visão	67
3.10	Interação com o Microcontrolador.....	69
3.10.1	Controlo da Máquina LV600 PRO	70
3.10.1	Sistema de locomoção	71
3.11	Deteção de Obstáculos com o LIDAR	73
3.12	Arquitetura de Navegação	74

3.12	Funcionamento Prático do Sistema Sentry	75
CAPÍTULO 4: Discussão de Resultados		77
4.1	Teleoperação	77
4.2	Teste da Orientação do Robô	78
4.2.1	Desempenho da Calibração	78
4.2.2	Desempenho da Filtragem	80
4.2.3	Desempenho do <i>Tilt</i>	92
4.3	Teste da Posição do Robô	94
4.3	Teste ao Algoritmo Desenvolvido para o Sistema de Visão.....	98
4.4	Teste ao Algoritmo do LIDAR.....	100
CAPÍTULO 5: Conclusões.....		102
5.1	Trabalho Futuro.....	103
REFERÊNCIAS BIBLIOGRÁFICAS		104
Anexo A		111
Anexo B		111
Anexo C		112
Anexo D		112
Anexo E		114
Anexo E		117
Anexo F.....		120
Anexo G		121

INDICE DE FIGURAS

Figura 1 – Área afetada por incêndios em países europeus (dados do EFFIS) [2].....	1
Figura 2 – Logotipo do projeto de investigação F4F	3
Figura 3 – MDB LV600 PRO.....	5
Figura 4 – Robot Shakey [12]	8
Figura 5 – Exemplo de um AGV	9
Figura 6 – Exemplo de um AMR	9
Figura 7 – Robot BACCHUS da Robotnik	11
Figura 8 – Robô Ranger da Ingeniarius	12
Figura 9 – Definição da posição do robô num sistema referencial	17
Figura 10 – Representação da posição de um robô diferencial.....	17
Figura 11 – Sistemas de posicionamento GPS, Glonass, Galileo, Beidou [25]	18
Figura 12 – Funcionamento dos Recetores GNSS [26].....	19
Figura 13 – Processo Navegação INS	20
Figura 14 – Funcionamento Acelerómetro [30]	21
Figura 15 – Magnetómetro efeito de Hall [31]	22
Figura 16 – Magnetómetro fluxgate [32].....	23
Figura 17 – Exemplo de um IMU	23
Figura 18 – Sensores Ultrassónicos	24
Figura 19 – Exemplo de LIDAR existentes no mercado	24
Figura 20 – Exemplo de uma Point Cloud fornecida por um sensor LIDAR	26
Figura 21 – Exemplo de um sensor GPS.....	27
Figura 22 – Erros de Multipercurso e Shadowing.....	28
Figura 23 – Localização do Equador e do Meridiano de Greenwich.....	29
Figura 24 – Domínio do tempo vs Domínio de frequência [41].....	31
Figura 25 – Exemplo de uma estrutura de um filtro IIR	31
Figura 26 – Gráfico que mostra o comportamento de ordens dadas a um filtro IIR [42]	33
Figura 27 – Exemplo de uma estrutura de um filtro IIR	33
Figura 28 – Resposta frequência de um filtro Passa-Baixo [43]	35
Figura 29 – Resposta frequência de um filtro Passa-Alto [43].....	35
Figura 30 – Resposta em frequência de um filtro passa-banda [43].....	36
Figura 31 – Resposta em frequência de um filtro rejeição de banda [43]	36
Figura 32 – Tipos de filtros Moving Average	37
Figura 33 – Funcionamento do Complementary Filter.....	39
Figura 34 – Funcionamento da visão computacional [49]	41
Figura 35 – Estrutura do Algoritmo YOLOv5. (Baseado em [52]).....	42
Figura 36 – Sistema de arquivos do ROS. (Baseado em [54])	44
Figura 37 – Funcionamento ROScore. (Baseada em [55]).....	44
Figura 38 – Exemplo da comunicação de um tópico [56]	45
Figura 39 – Partes constituintes do Sentry.....	46
Figura 40 – Final design do Sentry	46
Figura 41 – Integração Mecânica do Sentry na LV 600 PRO: (a) Verificação dos Campos de Visão das Câmaras; (b) Validação da Estrutura Mecânica na LV 600 PRO	47
Figura 42 – Método de operação da máquina utilizando o controlo remoto	48
Figura 43 – Arquitetura de Hardware do Sentry	48

Figura 44 – Minicomputadores escolhidos para o Sentry. a): Nvidia Jetson Xavier Nx; b): Intel NUC 1185G7	49
Figura 45 – Routers 3G/4G escolhidos para o Sentry: (a) RUT360; (b) D-link DWR-921	50
Figura 46 – Arquitetura de sensores do Sentry	53
Figura 47 – Duro Inertial	53
Figura 48 – Demonstração de todos os sensores montados no Sentry	55
Figura 49 – Princípio funcionamento do Sistema RTK com o Sentry	56
Figura 50 – Diagrama de blocos para obtenção do Heading.....	57
Figura 51 – Determinação do Heading com um magnetómetro de 3 eixos	58
Figura 52 – Processo para a calibração do magnetómetro	60
Figura 53 – Processo para aplicação de um LPF aos dados calibrados do magnetómetro..	61
Figura 54 – Exemplo funcionamento <i>deque</i> para o EMAF	62
Figura 55 – Diagrama de blocos para aplicação do EMAF	62
Figura 56 – Aplicação de um EMAF aos dados calibrados do magnetómetro.....	63
Figura 57 – Frames Duro em relação ao Acelerómetro, Giroscópio e Magnetómetro	63
Figura 58 – Modelo de posição MDB LV600 PRO	65
Figura 59 – Diagrama de blocos para obter posição filtrada	66
Figura 60 – Funcionamento do Filtro de Kalman onde existe a fusão entre o GPS e IMU para obter a posição	67
Figura 61 – Modelo do sistema de visão.....	68
Figura 62 – Processo para deteção de pessoas com distância usando o YOLOv5.....	69
Figura 63 – Máquina de estados para o controlo da LV600 PRO	70
Figura 64 – Método locomoção primeira abordagem. (a): locomoção do trator numa trajetória retilínea; (b): Movimento rotacional do trator no seu próprio eixo.....	72
Figura 65 – Movimentos combinados. (a): Movimento combinado onde a lagarta esquerda tem mais força que a direita; (b): Movimento combinado onde a lagarta direita tem mais força que a esquerda	73
Figura 66 – Arquitetura Final do Sistema Sentry para incorporar Algoritmos de Navegação	75
Figura 67 – Local de testes experimentais.....	77
Figura 68 – Interação entre o ROS e a MDB LV600 PRO.....	77
Figura 69 – Comparação dados calibrados vs dados descalibrados.....	78
Figura 70 – Pixhawk Px4 e Duro Inertial em paralelo para aferir o Heading.....	79
Figura 71 – Teste do Este magnético com o Eixo X da frame do Duro.	79
Figura 72 – Média do Máximo do Erro Absoluto no teste em regime estacionário com o LPF	81
Figura 73 – Média da Média do Erro Absoluto resultante do teste em regime estacionário com o LPF	81
Figura 74 – Média da Variância resultante do teste em regime estacionário com o LPF.....	81
Figura 75 – Resposta temporal do LP Filter (ordem 2) referente ao eixo X.....	82
Figura 76 – Resposta temporal do LPF (ordem 2) referente ao eixo Y	82
Figura 77 – Dados Filtrados em regime estacionário com o LPF (ordem 2) referente ao eixo Z	82
Figura 78 – Dados Filtrados em regime estacionário com o LPF (ordem 2) referente ao Heading	83
Figura 79 – Resposta temporal em regime de transição do LPF (ordem 2) referente ao eixo X.....	83

Figura 80 – Resposta temporal em regime de transição do LPF (ordem 2) referente ao eixo Y	83
Figura 81 – Resposta temporal em regime de transição do LPF (ordem 2) referente ao eixo Z	84
Figura 82 – Resposta temporal em regime de transição do LPF (ordem 2) referente ao Heading	84
Figura 83 – Média do Máximo do Erro Absoluto resultante do teste em regime estacionário com o EMAF	85
Figura 84 – Média da Média do Erro Absoluto resultante do teste em regime estacionário com o EMAF	85
Figura 85 – Média da Variância resultante do teste em regime estacionário com o EMAF ..	85
Figura 86 – Resposta temporal do EMAF referente ao eixo X em regime estacionário	86
Figura 87 – Resposta temporal do EMAF referente ao eixo Y em regime estacionário	86
Figura 88 – Resposta temporal do EMAF referente ao eixo Z em regime estacionário	86
Figura 89 – Resposta temporal do EMAF referente ao Heading em regime estacionário	86
Figura 90 – Resposta temporal em regime de transição do EMAF referente ao eixo X	87
Figura 91 – Resposta temporal em regime de transição do EMAF referente ao eixo Y	87
Figura 92 – Resposta temporal em regime de transição do EMAF referente ao eixo Z	87
Figura 93 – Resposta temporal em regime de transição do EMAF referente ao Heading	88
Figura 94 – Média do Máximo do Erro Absoluto resultante do teste em regime estacionário com o LPF e com o EMAF	88
Figura 95 – Média da Média do Erro Absoluto resultante do teste em regime estacionário com o LPF e com o EMAF	89
Figura 96 – Média da Variância resultante do teste em regime estacionário com o LPF e com o EMAF	89
Figura 97 – Resposta temporal em regime de transição do LPF e do EMAF referente ao eixo X	89
Figura 98 – Resposta temporal em regime de transição do LPF e do EMAF referente ao eixo Y	90
Figura 99 – Resposta temporal em regime de transição do LPF e do EMAF referente ao eixo Z	90
Figura 100 – Resposta temporal em regime de transição do LPF e do EMAF referente ao Heading	90
Figura 101 – Resposta temporal em regime de transição do LPF e do EMA no eixo X	91
Figura 102 – Resposta temporal em regime de transição do LPF e do EMA no eixo Y	91
Figura 103 – Resposta temporal em regime de transição do LPF e do EMA no eixo Z	91
Figura 104 – Resposta temporal em regime de transição do LPF e do EMA no Heading	91
Figura 105 – Estrutura para testes do <i>tilt</i> no laboratório	92
Figura 106 – Resultado da experiência do tilt no laboratório. (a): Movimento executado no eixo X; (b): Movimento executado no eixo Y; (c): Seta designada como a direção dada pelo Heading	92
Figura 107 – Teste do Tilt. (a): Ponto inicial do robô (b) Trajetória do robô com inclinação. (c) Fim do percurso	93
Figura 108 – Resposta temporal do Tilt durante uma trajetória em linha reta	94
Figura 109 – Teste A. (a): estrutura móvel; (b): estrutura fixa; (c): execução do teste	94
Figura 110 – Gráfico de barras comparativo dos resultados obtidos do teste A	95
Figura 111 – Pontos coletados durante a trajetória do Teste A	95

Figura 112 – Teste B realizado no terreno.....	96
Figura 113 – Gráfico de barras comparativo dos resultados obtidos do teste B.....	97
Figura 114 – Pontos coletados durante a trajetória do robô sem filtros e com o Filtro de Kalman e o Filtro SMA.....	97
Figura 115 – Observação dos erros presentes entre o filtro de Kalman e o não uso de filtro.....	97
Figura 116 – Teste para verificar o funcionamento do YOLOv5. (a): <i>setup</i> com duas câmaras térmicas, uma câmara RGB e uma câmara Depth; (b): resposta do YOLOv5.....	99
Figura 117 – Distâncias definidas para comparação dos dados reais com os dados da câmara	99
Figura 118 – Procedimento de uma das etapas do teste do Algoritmo do sistema de visão no terreno. Neste caso a pessoa situava-se a 6 metros da câmara frontal do Sentry	99
Figura 119 – Algoritmo Adaptive clustering na point cloud do Velodyne VLP-16.....	101

ÍNDICE DE QUADROS

Tabela 1 – Características MDB LV600 PRO	5
Tabela 2 – Comparação LIDAR 2D vs LIDAR 3D	25
Tabela 3 – Package ROS para comunicar com os sensores/componentes do Sentry	56
Tabela 4 – Valores definidos no Microcontrolador	71
Tabela 5 – Valores Máximo e RMS obtidos durante a trajetória do teste A.....	95
Tabela 6 – Dados obtidos da experiência no terreno	96
Tabela 7 – Valores obtidos pelo teste executado no sistema de visão.....	100

SIMBOLOGIA E ABREVIATURAS

ADAI	Associação para o Desenvolvimento da Aerodinâmica Industrial
AGV	<i>Automated Guided Vehicle</i>
AMR	<i>Autonomous Mobile Robot</i>
CCI	Centro Comum de Investigação
CMA	<i>Cumulative Moving Average</i>
EFFIS	<i>European Forest Fire Information System</i>
EKF	<i>Extended Kalman Filter</i>
EMAF	<i>Exponential Moving Average Filter</i>
F4F	<i>Forest for the Future</i>
FCTUC	Faculdade de Ciências e Tecnologia da Universidade de Coimbra
FIR	<i>Finite Impulse Response</i>
FOV	<i>Field of View</i>
GNSS	<i>Global Navigation Satellite System</i>
GPS	<i>Global Positioning System</i>
IIR	<i>Infinite Impulse Response</i>
IMU	<i>Inertial Measurement Unit</i>
INS	<i>Inertial Navigation System</i>
ISEC	Instituto Superior de Engenharia de Coimbra
KF	<i>Kalman Filter</i>
LIDAR	<i>Light Detection And Ranging</i>
LLT	<i>Low Level Trigger</i>
LPF	<i>Low-Pass Filter</i>
MEMS	<i>Micro Electromechanical Systems</i>
ML	<i>Machine Learning</i>
RGB	<i>Red, Green and Blue</i>
RMS	<i>Root Mean Square</i>
ROS	<i>Robot Operating System</i>
RTK	<i>Real-Time Kinematic</i>
SMA	<i>Simple Moving Average</i>
SMAF	<i>Simple Moving Average Filter</i>

TOA	<i>Time of Arrival</i>
TOF	<i>Time-of-Flight</i>
UAV	<i>Unmanned Aerial Vehicle</i>
UTM	<i>Universal Transverse Mercator</i>
WMA	<i>Weighted Moving Average</i>
YOLO	<i>You Only Look Once</i>

CAPÍTULO 1: Introdução

Os incêndios florestais destruíram, nestes últimos anos, vastas áreas florestais em diferentes partes do mundo, causando impactos negativos na natureza tais como: poluição da água, perda de vegetação, desaparecimento da fauna e a diminuição da qualidade do ar [1]. A Figura 1 representa uma série temporal disponibilizada pelo Sistema Europeu de Informação sobre Incêndios Florestais, *European Forest Fire Information System* (EFFIS), relativamente a estudos desenvolvidos pelo Centro Comum de Investigação (CCI), e reporta o número de incêndios, bem como a área queimada desde 1980 até à data, no sul da França, Grécia, Itália, Portugal e Espanha (EUMED5). Da análise dos valores ao longo deste período, verifica-se que houve uma ligeira diminuição na área queimada no EUMED5, com exceção de Portugal. No entanto, esta tendência é fortemente influenciada pela variabilidade interanual, que está diretamente relacionada com as condições meteorológicas. Um exemplo foi em 2017, em que a área queimada foi a segunda maior e sem precedentes em Portugal. No ano seguinte, a área queimada foi a menor registada nesta série temporal [2].

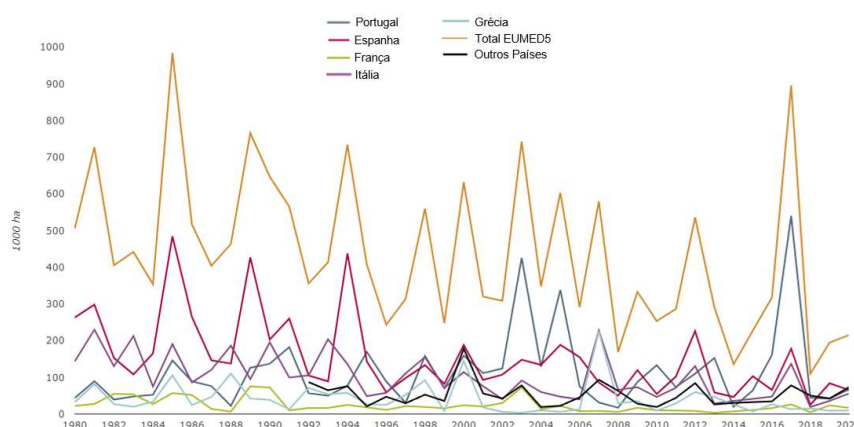


Figura 1 – Área afetada por incêndios em países europeus (dados do EFFIS) [2]

Os incêndios florestais têm consequências negativas na economia, causando a perda de recursos primários provenientes da floresta, conforme mencionado por Ribeiro et al. em [3]. Além disso, os incêndios florestais provocam o abandono das comunidades rurais, resultando numa menor vigilância das áreas florestais, diminuição de empregos e impacto negativo no turismo [3]. Para mitigar os incêndios florestais, é essencial implementar medidas para remover e limpar a vegetação, sendo essas ações cruciais para reduzir efetivamente o risco de incêndio [4]. Assim sendo, anualmente são estabelecidas novas regulamentações e campanhas de conscientização, que obrigam os proprietários, arrendatários, ou entidades que detenham terrenos em espaços rurais e urbanos a proceder à sua limpeza, a fim de prevenir ou reduzir o risco de incêndio [3].

A limpeza de terrenos compreende essencialmente o corte de ervas, arbustos, mato, entre outros materiais vegetais, numa faixa com largura não inferior a 50 metros em torno dos edifícios localizados em áreas rurais ou florestais e a limpeza dos sobrantes após a limpeza.

O presente trabalho descreve o desenvolvimento de uma plataforma robótica florestal autónoma para realizar o corte da vegetação para a prevenção de incêndios, onde o foco principal centra-se na integração de um sistema sensorial para fornecer dados com robustez para a navegação autónoma da plataforma robótica.

Neste primeiro capítulo vai ser analisado o processo que está por detrás do desenvolvimento deste trabalho, realçando a prevenção de incêndios e a procura de recursos tecnológicos que permitam promover e beneficiar a prevenção de incêndios. De seguida, é apresentado o enquadramento deste trabalho perante as entidades financiadoras, bem como a descrição dos objetivos da máquina florestal que foi utilizada como plataforma robótica. Finalmente, apresenta-se a estrutura deste relatório de projeto, bem como uma descrição sobre cada capítulo.

1.1 Motivação

O início de um incêndio florestal começa antes do próprio incêndio em si, e uma das causas é a acumulação da biomassa vegetal disponível para a sua combustão, sendo, desta forma, crucial implementar medidas para mitigar a sua ocorrência.

Uma das medidas a implementar envolve a necessidade de cortar e ou remover a vegetação em áreas arborizadas, o que efetivamente reduz o risco de incêndios [3]. No entanto, atualmente, já existem métodos para a remoção da vegetação por meio de máquinas trituradoras, que se têm revelado uma mais-valia na limpeza de áreas florestais [4].

Estas máquinas são operadas de forma manual, com um operador designado para cada máquina. Este pode estar presente na própria máquina ou controlar remotamente a sua operação por meio de comandos emitidos em sinais rádio, os quais manipulam a máquina. A operação deste tipo de máquinas é extremamente perigosa e requer um elevado grau de responsabilidade, além de habilidades técnicas especializadas, dadas as suas grandes dimensões, com o consequente potencial de causar graves acidentes [5]. Portanto, torna-se imperativo implementar recursos tecnológicos que permitam aos operadores trabalhar com segurança, minimizando os riscos associados a este tipo de máquina.

Atualmente, poucos projetos de investigação estão focados no desenvolvimento de recursos tecnológicos para prevenir incêndios florestais, o que destaca a necessidade de realizar este trabalho.

1.2 Enquadramento

A Associação para o Desenvolvimento da Aerodinâmica Industrial (ADAI), é um Centro Tecnológico vinculado ao Departamento de Engenharia Mecânica da Faculdade de Ciências e Tecnologia da Universidade de Coimbra (FCTUC), onde tem participado ultimamente em projetos de investigação, dedicados ao desenvolvimento de sistemas autónomos para o combate de incêndios florestais. Refira-se que a ADAI é uma entidade de direito privado sem fins lucrativos que se dedica à realização de atividades de investigação, desenvolvimento, formação e prestação de serviços. O presente projeto foi realizado nas instalações da ADAI enquadrado nas diretrizes científicas e sob orientação de docentes do Instituto Superior de Engenharia de Coimbra (ISEC).

Este projeto enquadra-se também no projeto piloto PP8 - Demonstração das Técnicas e Ferramentas de Limpeza Florestal, do projeto F4F- *Forest for the Future* (CENTRO-08-5864-FSE-000031) cujo o logotipo se apresenta na Figura 2, tendo como principal objetivo desenvolver e demonstrar as capacidades das máquinas florestais autónomas para o corte da vegetação espontânea. Para alcançar esse objetivo, são adaptadas plataformas florestais telecomandadas já existentes no mercado, com a instalação de *hardware* e *software* de controlo necessários para a sua localização e operação, de maneira segura e eficiente e soluções inovadoras para máquinas florestais para o corte de vegetação. No que diz respeito ao *hardware*, foram instaladas câmaras RGBD e de infravermelhos, *Light Detection And Ranging* (LIDAR), *Global Positioning System* (GPS), e diversos outros sensores tais como, acelerómetros e giroscópios, bem como, toda a eletrónica de controlo associada, baterias e suportes para integração, tendo sido necessárias adaptações na estrutura da máquina [6].



Figura 2 – Logotipo do projeto de investigação F4F

O desenvolvimento deste projeto foi dividido em diferentes tarefas onde, para além do autor deste projeto, colaboraram os alunos Tiago Gameiro do ISEC e o aluno Francesco Di Giorgio do Politécnico de Torino.

1.3 Objetivos

O principal objetivo deste trabalho centrou-se no desenvolvimento de um sistema sensorial, designado por Sentry, que se adapte a uma Máquina Trituradora Florestal a fim de a tornar autónoma. Nesse processo assume particular relevância a implementação de uma metodologia que permita adquirir e processar um conjunto de dados os quais serão aplicados em algoritmos de navegação para a operação da plataforma robótica em modo autónomo e desta maneira efetuar o estudo e avaliação dos filtros e algoritmos propostos. O trabalho realizado foi desenvolvido em torno da máquina trituradora / plataforma robótica MDB LV600 PRO.

Portanto, para definir e validar o desenvolvimento deste sistema sensorial, vários passos foram executados a saber:

- Estudo e compreensão das ações eletromecânicas da máquina florestal;
- Definição do *hardware* necessário, a colocar para a locomoção da máquina e para a avaliação do seu comportamento quando estiver em navegação, como por exemplo a posição, orientação, etc.;
- Implementação de uma metodologia que permita fornecer dados, a fim de se proceder à aplicação em algoritmos de navegação;
- Estudo de algoritmos de filtragem e calibração para a remoção de ruído dos sensores e para a sua calibração;
- Avaliação do impacto de cada sensor na máquina florestal, bem como os filtros e calibração propostos.

1.4 Descrição da Máquina Florestal

A máquina florestal utilizada neste projeto é a Green Climber LV 600 PRO da MDB, representada na Figura 3, um trator controlado remotamente, projetado especialmente para operar em áreas perigosas, como por exemplo, terrenos inclinados até 60 graus. Esta máquina é uma escolha ideal para realizar tarefas de limpeza da vegetação de estradas e rodovias, mas também de florestas com alta intensidade de vegetação. A referida máquina oferece versatilidade e robustez para atender às necessidades específicas do trabalho em questão [7].



Figura 3 – MDB LV600 PRO

Esta máquina, possui as características descritas na Tabela 1, nomeadamente um baixo consumo de combustível, dimensões compactas e peso reduzido permitindo assim o seu transporte de forma mais fácil mesmo para zonas de acesso mais difícil.

Tabela 1 – Características MDB LV600 PRO

Propriedades	Parâmetros
Inclinação Máxima	60°
Transmissão	Hidráulica
Velocidade	0-9 km/h
Comprimento	1360 mm até 1760 mm
Tamanho com a Ferramenta	2160 mm + 750 mm
Peso com a Ferramenta	1292 kg + 227 kg
Potência	56 cv
Alcance controlo remoto	300 metros

A operação desta máquina é realizada através de um controlo remoto que envia sinais em rádio frequência para um módulo que está conectado às saídas do trator permitindo assim o controlo dos movimentos hidráulicos dos equipamentos instalados na máquina. As saídas da máquina correspondem por exemplo, o movimento das lagartas, a regulação da altura da ferramenta, o controlo das rotações do motor, o controlo do triturador, entre outras. Estas saídas estão associadas a uma ficha de 64 pinos, estando cada pino associado a uma ação da máquina.

1.5 Estrutura do Projeto

Nos capítulos seguintes deste projeto serão apresentadas as restantes etapas do trabalho desenvolvido:

- No segundo capítulo apresenta-se o estado-da-arte relativamente à robótica, nomeadamente a evolução, bem como o estado da arte nas áreas científicas envolvidas neste tipo de trabalhos;
- No terceiro capítulo, apresenta-se a metodologia proposta, bem como todas as técnicas e algoritmos utilizados para o desenvolvimento do sistema sensorial;
- No quarto capítulo são apresentados os resultados obtidos com base nos testes realizados, de forma a validar a metodologia proposta;
- No quinto capítulo são reveladas as conclusões sobre o trabalho desenvolvido e uma perspetiva sobre o trabalho futuro.

CAPÍTULO 2: Estado da Arte

Este capítulo, pretende apresentar uma breve introdução sobre a robótica móvel, a sua evolução, e as suas aplicações nomeadamente nas áreas florestal e agrícola. De seguida, são descritos os sistemas de localização existentes, indicando a distinção entre a localização absoluta e relativa. São também analisadas as formas de orientação e a posição dos robôs móveis, bem como os diferentes sistemas de navegação para robôs móveis. São apresentados os sensores utilizados neste tipo de robôs, os sistemas de coordenadas GPS e os filtros digitais aplicados para eliminar o ruído dos dados recolhidos dos sensores. As metodologias de fusão sensorial, bem como o contributo da visão computacional na área da robótica móvel são também analisadas. Por fim, descreve-se o funcionamento da *framework* ROS e as vantagens de a integrar no desenvolvimento de aplicações para robôs móveis.

2.1 Robótica

A robótica tem tido uma ampla gama de aplicações em várias áreas, desde a indústria, medicina, agricultura e na floresta. Não existe nenhuma designação geral sobre o termo robótica, contudo, conforme a *Encyclopedia Britannica, Inc.*, um robô consiste numa máquina que realiza tarefas de forma autónoma, substituindo assim o esforço humano [8].

Este termo surgiu pela primeira vez em 1921, numa peça de teatro do escritor checo Karel Capek. No entanto, somente em 1954 foi desenvolvido nos EUA o primeiro robô industrial, designado por Unimate, que consistia num braço hidráulico utilizado na linha de montagem da *General Motors*, com a finalidade de levantar cargas pesadas. Nos anos seguintes foram desenvolvidas alterações a este robô que permitiram a realização de outro tipo de operações [9].

2.1.1 Robôs Móveis

A partir dos robôs manipuladores industriais, surgiram os robôs móveis pelas limitações apresentadas de falta de mobilidade dos robôs industriais. Um robô móvel pode tomar decisões em ambientes não estruturados, sem a necessidade de um operador para os supervisionar [10].

Este ramo da robótica tem vindo a tornar-se um dos principais focos de atenção da robótica, sendo atualmente um ramo onde se desenvolve mais investigação, tendo como objetivo superar as limitações dos manipuladores através da inclusão da inteligência artificial, conduzindo ao desenvolvimento de inúmeras aplicações, dos robôs móveis, em diferentes áreas [10].

Um dos primeiros robôs móveis produzidos, foi em 1969 na Universidade de Stanford, em que se apresentou o robô móvel Shakey, como se pode observar na Figura 4. Este robô foi o primeiro a utilizar inteligência artificial para controlar os seus

movimentos, incorporando sensores, tais como uma câmara e sensores táteis para o efeito. A sua navegação ocorria em ambientes estruturados, como salas de laboratórios, escritórios. O seu processamento era realizado em dois computadores distintos localizados, um a bordo e outro externamente, os quais comunicavam via rádio [11].



Figura 4 – Robot Shakey [12]

Após o desenvolvimento deste robô, os trabalhos de investigação neste ramo da robótica intensificaram-se ainda mais. Na década de 70 foi desenvolvido pela NASA um robô designado por Mars-Rover com a finalidade de explorar terrenos hostis e desconhecidos. Este robô consistia numa plataforma que integrava um braço mecânico, sensores de proximidade, câmaras e sensores laser [10].

O aumento da capacidade de processamento dos microcontroladores e computadores facilitou e tornou mais eficiente lidar com grandes quantidades de informação e dados, o que tornou a área da robótica ainda mais dinâmica e, conseqüentemente começou a apresentar cada vez mais potencialidades tornando-se assim mais capaz de se adaptar num determinado ambiente sem supervisão externa.

Atualmente os robôs móveis são cada vez mais comuns e têm tido um enorme impacto na indústria e em outros sectores. Estes são desenvolvidos para navegar de forma autónoma em diferentes cenários/ambientes, adequando-se às exigências impostas pelas diferentes aplicações [12].

Os dois tipos mais comuns de robôs móveis são os *Automated Guided Vehicles* (AGVs) e os *Autonomous Mobile Robots* (AMRs) [12, 13].

Os AGVs seguem rotas ou trajetórias estabelecidas, onde se movem ao longo de um percurso pré-estabelecido, utilizando várias técnicas de orientação, tais como sensores magnéticos, marcadores no chão, lasers ou sistemas de visão. Normalmente, este tipo de robôs é utilizado em ambientes industriais, como fábricas e armazéns, como se pode analisar na Figura 5, de forma a movimentar materiais ou produtos automaticamente de um lugar para outro. No entanto, estes tipos de robôs operam em ambientes controlados, com caminhos e pontos pré-estabelecidos [13].



Figura 5 – Exemplo de um AGV

Em contrapartida, os AMRs têm a autonomia de navegar e tomar decisões independentes em tempo real. Estes não necessitam de uma rota predefinida, e desta forma conseguem-se adaptar a ambientes dinâmicos. Este tipo de robô utiliza uma vasta gama de sensores, como LIDAR, ultrassons, câmaras, de forma a que consigam desviar-se de obstáculos e planejar autonomamente as próprias rotas. Por outro lado, os AMRs têm uma ampla gama de aplicações, desde a saúde, logística, indústria, mas também apresentam flexibilidade e capacidade de adaptação, como se pode observar na Figura 6 [13].



Figura 6 – Exemplo de um AMR

Em suma, a principal diferença entre estes dois tipos de robôs está na forma como navegam, ou seja, um AMR consegue ter um maior nível de autonomia que um AGV, devido ao facto de conseguir adaptar em ambientes não estruturados. A plataforma robótica deste trabalho segue uma tipologia do tipo AMR e assim sendo, neste trabalho abordam-se os AMRs, que apresentam as seguintes vantagens:

- Segurança para os trabalhadores, já que em muitas atividades existem tarefas que são impossíveis e/ou perigosas para o ser humano realizar [12];
- Aumento da eficiência, pois permite que as tarefas sejam distribuídas e desta forma aumentam a produtividade e reduzem os custos [12];
- Flexibilidade, ou seja, um ARM pode ser ajustado a várias tarefas e sem a supervisão constante de um operador [12].

As aplicações para os robôs móveis são inúmeras, de entre elas destacam-se a integração de ARMs em armazéns e fábricas. Estima-se que os trabalhadores usam cerca de 60% do seu tempo a realizar a deslocação de materiais para repor linhas de produção, organizar o armazém, entre outros tipos de tarefas, sendo certo que muitas destas atividades rotineiras e repetitivas poderão ser executadas por um robô móvel aumentando assim a produtividade e ao mesmo tempo libertando os trabalhadores para outras tarefas. [12].

Neste campo, os ARMs podem executar os seguintes tipos de operações:

- *Order picking*;
- Transporte de material dentro das instalações;
- Separação de peças e embalagens;
- Entrega de produtos em áreas de expedição para serem enviados aos consumidores.

Por outro lado, também existem ARMs incorporados com braços robóticos colaborativos que podem executar operações, tais como:

- Paletização;
- Empacotamento;
- Inspeção de qualidade em diferentes pontos da linha.

2.1.2 Robótica na Agricultura e na Floresta

Uma outra aplicação compreende a integração de robôs na floresta e na agricultura, o que apresenta um impacto substancial [14]. Os agricultores estão cada vez mais conscientes que necessitam de veículos modernos para os auxiliar nos trabalhos agrícolas, dada a escassez de mão-de-obra, o aumento de procura de produtos, e o aumento dos custos de produção, que levaram à automatização de tarefas de forma a reduzir custos.

Assim sendo, a inclusão da robótica na agricultura tem como objetivo aumentar a produtividade, eficiência e a sustentabilidade ambiental deste sector [14].

As principais aplicações dos robôs na agricultura compreendem [15]:

- Pulverização com produtos químicos, conforme exigido por cada fruta ou planta;
- Colheita de frutas ou vegetais, através de uma plataforma robótica incorporada num braço robótico colaborativo;
- Paletização;
- Recolha de dados sobre o ambiente agrícola como o estado de plantas, solo, humidade, etc.;
- Seleção correta dos alimentos de forma a evitar o desperdício.

Um exemplo de um projeto associado à robótica na agricultura é o sistema robótico móvel BACCHUS, presente na Figura 7 e que pretende reproduzir as operações de colheita manual substituindo-se às pessoas [15], operando em vários níveis diferentes como:

- Durante a navegação autónoma o robô inspeciona e recolhe dados acerca do terreno agrícola através do seu sistema de sensores;
- O robô realiza operações de colheita utilizando dois braços robóticos e com o cuidado necessário conforme se mostra na Figura 7;
- A garra do braço robótico consegue ajustar-se perante a geometria do fruto a recolher.



Figura 7 – Robot BACCHUS da Robotnik

O futuro da robótica na agricultura é tópico empolgante. Os novos desenvolvimentos e avanços tecnológicos garantem que a complexidade e a inteligência dos robôs agrícolas os tornam cada vez mais eficientes e inteligentes na execução de tarefas agrícolas [16]. Os robôs agrícolas estão a tornar-se mais dominantes em todo o mundo, especialmente à medida que a precisão do seu trabalho e sua capacidade de aprendizagem acomoda a complexidade incrementando a sua robustez e eficácia. Isso não significa que os trabalhadores humanos perderão empregos, mas sim que vão ter mais meios para realizar as suas tarefas com mais eficiência e com menos esforço. As economias avançadas já têm dificuldades em preencher cargos agrícolas, devido ao esforço físico que certas tarefas exigem e no futuro será também necessário empregar trabalhadores altamente qualificados em funções de manutenção, reparação e supervisão dos robôs que executam este tipo de tarefas agrícolas [16]. Exemplos de possíveis desenvolvimentos futuros na robótica agrícola incluem o uso da tecnologia tais como drones automatizados para plantar culturas em áreas de terras agrícolas extensas, ou drones para regar plantações em períodos de seca [16].

A utilização da robótica na floresta pode abranger diversas aplicações, no entanto a mais relevante compreende a limpeza e remoção de biomassa vegetal através de ferramentas que estejam acopladas. Como referido anteriormente na secção 1.2, ao realizar a remoção deste material inflamável na floresta é possível reduzir a

incidência e intensidade dos incêndios florestais. Dessa forma é evidente a necessidade de se efetuarem avanços na área da robótica florestal apesar do seu desenvolvimento se encontrar numa fase inicial. De entre aos desafios a ultrapassar refira-se:

- A utilização em ambientes com terrenos não estruturados [17];
- A utilização de sistemas para a navegação em ambientes florestais [18];
- As limitações da utilização de GPS pela existência de vegetação mais densa que representa um problema na localização e mapeamento do robô [19].

Um exemplo de um projeto associado à robótica autónoma na floresta é o projeto SEMFIRE que foi criado com o objetivo de reduzir a biomassa nas florestas. O sistema consiste em vários robôs, incluindo um grupo de pequenos *Unmanned Aerial Vehicles* (UAVs) chamados Scouts, que fazem um varrimento numa área para identificar regiões de interesse. Essas informações são então transmitidas ao robô Ranger como se mostra na Figura 8 que é a plataforma responsável pela tarefa de limpeza [20].



Figura 8 – Robô Ranger da Ingeniarius

2.2 Sistemas de Localização

A localização é um aspeto fundamental para a operação de qualquer robô autónomo. Para que um robô opere autonomamente no seu espaço de trabalho, este deve-se integrar com um sistema de localização/posicionamento para estimar a posição e ângulo de direção do robô. No entanto, a localização num dado ambiente depende do tipo de mapa utilizado para representar a sua localização. Existem dois tipos principais de mapas: os mapas métricos e os mapas topológicos. Os mapas métricos utilizam medidas quantitativas do ambiente, permitindo determinar a posição do veículo em relação a um sistema de coordenadas, sendo a representação *Universal Transverse Mercator* (UTM), que irá ser abordada na secção 2., a mais comum. Já os mapas topológicos utilizam marcos (*landmarks*) detetados por sensores de visão para descrever o ambiente, sendo mais utilizados em ambientes internos, já que em ambientes externos a representação se torna mais difícil pela ausência de padrões [21].

2.2.1 Localização Absoluta

Para obter a localização absoluta de um robô é necessário saber o seu posicionamento global. Há diversos meios para obter a localização absoluta, sendo um deles a detecção visual recorrendo a *landmarks*. Nesta técnica, as marcas de posições globais já conhecidas são usadas como referência, e a detecção dessas posições e das distâncias é feita para se obter a localização. A detecção das referidas marcas é a primeira dificuldade nesta abordagem, onde a visão computacional é a principal tecnologia utilizada, porém esta pode ser uma desvantagem até porque é necessário ter o conhecimento prévio de marcas existentes no local de trabalho do robô [21].

O recetor GPS tornou-se o equipamento favorito para a obtenção da localização absoluta, como será analisado com mais detalhe na secção 2.5. No entanto, o GPS como todos os sensores apresenta algumas limitações, e para que este funcione corretamente é necessário que a antena esteja apontada para céu aberto [21].

2.2.2 Localização Relativa

A localização relativa baseia-se num ponto de partida conhecido, a partir do qual as medidas são integradas para se obter as mudanças de posição. Existem dois tipos principais de localização relativa: uma delas é a odometria e a outra a navegação inercial [21].

A odometria é baseada na integração da medida da velocidade da roda do veículo ao longo do tempo para estimar a posição, mas também requer o conhecimento do ângulo de desvio do robô. No entanto, o somatório de erros ocorre à medida que a velocidade é integrada, existindo dois tipos de erros: os erros sistemáticos e erros não sistemáticos. Os erros sistemáticos estão relacionados ao modelo cinemático do veículo, das rodas e podem ser corrigidos com calibração, enquanto os erros não sistemáticos ocorrem quando há falta de contacto entre as rodas e o chão, deslizamentos ou algum deslocamento lateral [21].

Na navegação inercial conseguem-se obter medidas de aceleração e a velocidade angular através de sensores *Inertial Measurement Unit* (IMU), que serão abordados na secção 2.5, que possuem 3 eixos ortogonais utilizando um acelerómetro, giroscópico e magnetómetro. A obtenção da posição com recurso ao sensor IMU é uma solução mais barata, e não é afetada com o problema do deslizamento das rodas. No entanto existe um problema que é o facto da posição ser calculada integrando as leituras do IMU ao longo do tempo da navegação do robô e desta forma o problema surge devido à acumulação de erros durante a integração realizada ao longo percurso do robô [21].

2.3 Representação da Orientação em Robôs Móveis

Para descrever a posição e orientação de um robô, uma abordagem comum é utilizar um sistema de coordenadas tridimensional, em que os eixos representam as direções X , Y e Z e a orientação do robô pode ser descrita através de ângulos de Euler ou quaterniões, estes representam a rotação do robô em torno de cada um desses eixos.

2.3.1 Representação da Orientação

A obtenção da orientação θ do robô, pode ser feita utilizando diferentes abordagens. Neste relatório referem-se três métodos distintos sendo estes: a Matriz de Rotação; os Ângulos de Euler e o método dos Quaterniões.

a) Matriz de Rotação

A matriz de rotação é uma matriz 3×3 que oferece uma representação ortogonal usada para descrever as rotações de vetores e facilitar a conversão de coordenadas entre diferentes sistemas de referência. Para descrever de forma completa uma única rotação, geralmente precisamos de três ângulos distintos, frequentemente chamados de ângulos de Euler (γ , β , α). Portanto, quando desejamos girar um objeto em torno dos eixos X , Y ou Z , podemos utilizar a seguinte representação [22]:

$$Rot_x(\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\gamma) & -\sin(\gamma) \\ 0 & \sin(\gamma) & \cos(\gamma) \end{bmatrix} \quad (1)$$

$$Rot_y(\beta) = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \quad (2)$$

$$Rot_z(\alpha) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

b) Ângulos de Euler

Os ângulos de Euler constituem um conjunto de três ângulos utilizados para descrever a disposição de um corpo rígido em relação a um sistema de coordenadas que permanece fixo. Eles são utilizados para representar a orientação ou posição relativa de um referencial. Esses três ângulos denotam as rotações básicas ocorridas ao longo dos eixos X , Y e Z . Normalmente, são representados pelas notações γ , β e α , e suas definições são as seguintes [22]:

- γ refere-se à rotação do corpo em torno de seu eixo X fixo, também conhecida como "*roll*".
- β corresponde à rotação do corpo em torno de seu eixo Y fixo, também chamada de "*pitch*".
- α indica a rotação do corpo em torno de seu eixo Z fixo, também denominada "*yaw*" (ou orientação).

Usando a descrição de ângulos de Euler Z-Y-X, a matriz de rotação pode ser expressa da seguinte forma [22]:

$$R(\gamma, \beta, \alpha) = R_Z(\alpha) R_Y(\beta) R_X(\gamma) \quad (4)$$

E desta forma podemos reescrever a equação 4 desta forma:

$$R(\gamma, \beta, \alpha) = \begin{bmatrix} \cos(\alpha) \cos(\beta) & \cos(\alpha) \sin(\beta) \sin(\gamma) - \sin(\alpha) \cos(\gamma) & \cos(\alpha) \sin(\beta) \cos(\gamma) + \sin(\alpha) \sin(\gamma) \\ \sin(\alpha) \cos(\beta) & \sin(\alpha) \sin(\beta) \sin(\gamma) + \cos(\alpha) \cos(\gamma) & \sin(\alpha) \sin(\beta) \cos(\gamma) - \cos(\alpha) \sin(\gamma) \\ -\sin(\beta) & \cos(\beta) \sin(\gamma) & \cos(\beta) \cos(\gamma) \end{bmatrix} \quad (5)$$

Através de manipulação algébrica na equação 5, consegue-se representar os ângulos de Euler Z-X-Y da seguinte forma [22]:

$$\alpha = \text{atan2}(r_{21}, r_{11}) \quad (6)$$

$$\beta = \text{atan2}(-r_{31}, \sqrt{r_{32}^2 + r_{33}^2}) \quad (7)$$

$$\gamma = \text{atan2}(r_{32}, r_{33}) \quad (8)$$

No entanto é necessário ter em atenção as singularidades, como por exemplo em *pitch*, e desta forma asseguram-se as seguintes limitações [22]:

$$\begin{bmatrix} 0 \\ \frac{\pi}{2} \\ 0 \end{bmatrix} \leq \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} \leq \begin{bmatrix} 2\pi \\ \frac{\pi}{2} \\ 2\pi \end{bmatrix} \quad (9)$$

c) Quaterniões

A rotação em torno de um eixo também pode ser representada utilizando quaterniões. Os quaterniões consistem em duas partes principais: uma componente escalar, denotada como q_s , e uma componente vetorial, referida como q_v . Isso é descrito pela seguinte fórmula [22]:

$$q = (q_s, q_v) = q_4 + iq_1 + jq_2 + kq_3 \quad (10)$$

onde i, j e k são componentes do vetor $q_v = (q_1, q_2, q_3)$ e q_4 é a parte escalar $q_s = q_4$.

A partir de uma rotação em torno de um eixo, é viável calcular as componentes do quaternião por meio das seguintes equações [22]:

$$q_v = \sin\left(\frac{\Lambda}{2}\right) \begin{bmatrix} k_x \\ k_y \\ k_z \end{bmatrix} \quad (11)$$

$$q_s = \cos\left(\frac{\Lambda}{2}\right) \quad (12)$$

onde $\Lambda \in [0, \pi]$ é o ângulo de rotação e $K = [Kx, Ky, Kz]^T$ é o vetor de norma unitária que define o eixo de rotação. De forma notória, é importante destacar que um quaternião possui uma norma unitária, o que possibilita interpretá-lo como as coordenadas de um ponto numa esfera unitária de quatro dimensões.

A matriz de rotação pode ser obtida diretamente a partir do quaternião, conforme a seguinte relação:

$$R(q) = (q_s^2 - \|q_v\|^2) I_3 + 2q_v q_v^T + 2q_s (q_v) \quad (13)$$

Aprimorando a expressão 13, consegue-se obter a expressão do quaternião através dos elementos da matriz de rotação da seguinte forma [22]:

$$q_v = \frac{1}{4q_s} \begin{bmatrix} r_{23} - r_{32} \\ r_{31} - r_{13} \\ r_{12} - r_{21} \end{bmatrix} \quad (14)$$

$$q_s = \frac{1}{2} \sqrt{1 + r_{11} + r_{22} + r_{33}} \quad (15)$$

A representação por quaterniões apresenta vantagens pelo facto de não ser afetada por singularidades e apresenta ser mais eficiente a nível computacional [22].

2.3.2 Representação da Posição

Um robô opera dentro de um ambiente físico e suas ações, estão intrinsecamente ligadas à sua localização precisa no ambiente, incluindo tanto a posição como a orientação. Pode-se assumir que um robô é um corpo rígido onde as suas partes constituintes mantêm uma posição relativa constante entre si e o corpo rígido.

Para o efeito definiu-se um sistema de coordenadas sendo $\{W\}$ o ponto inicial da navegação do robô, com se pode observar na Figura 9. Supondo que este está no ponto de referência P , com um novo sistema de coordenadas $\{R\}$. Desta forma o vetor r representa o deslocamento entre sistema de coordenadas $\{W\}$ e o sistema de coordenadas $\{R\}$, presentes no ponto P . Este sistema de coordenadas tanto pode ser definido em coordenadas polares como em coordenadas cartesianas.

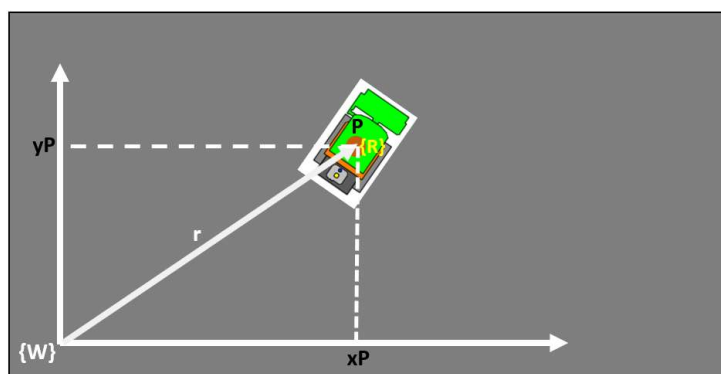


Figura 9 – Definição da posição do robô num sistema referencial

Então a posição do robô em relação a $\{W\}$ é descrita pela posição e orientação do sistema de coordenadas local $\{R\}$. A máquina florestal MDB LV600 PRO é do tipo diferencial, representado na Figura 10 sendo a sua configuração num sistema de coordenadas bidimensional definida pela:

1. Posição (x, y) ;
2. Ângulo de orientação θ , em relação aos eixos de coordenadas.

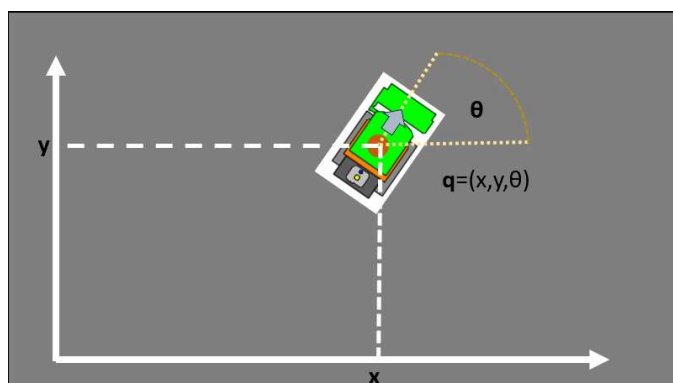


Figura 10 – Representação da posição de um robô diferencial

O vetor q representa as três componentes distintas da posição do robô.

2.4 Sistemas de Navegação

Pode-se dizer que a navegação “é o ato, atividade ou processo de encontrar o caminho para chegar de um lugar para outro” [23]. Atualmente com o avanço da tecnologia, a navegação tornou-se mais automatizada para os veículos. Existem vários tipos de navegação, no entanto na condução autónoma, salientam-se o *Global Navigation Satellite System* (GNSS) e o *Inertial Navigation System* (INS) [24].

2.4.1 Sistemas de Localização por Satélite

O sistema de navegação por satélite GNSS é baseado na transmissão de sinais de rádio pelos satélites em órbita, que são recebidos e interpretados pelos recetores GNSS, fornecendo informações precisas sobre posição, velocidade, tempo e direção/orientação onde o recetor se encontra. Esse sistema é capaz de fornecer

uma localização absoluta. O GNSS mais comum é o GPS, que é gratuito para todo o mundo, e fornecido pelo exército americano, porém outros países possuem os próprios GNSS, como por exemplo a Rússia possui o GLONASS e a China que está a desenvolver o COMPASS e a União Europeia o sistema Galileo. Estes sistemas permitem a cobertura para o mundo inteiro, como se pode observar na Figura 11 [24].



Figura 11 – Sistemas de posicionamento GPS, Glonass, Galileo, Beidou [25]

Estes sistemas funcionam com base no conceito de medições *Time of Arrival* (TOA), também conhecido como tempo de propagação, para determinar a posição do utilizador do sistema. Nos sistemas GNSS, esta medição envolve determinar o tempo que um sinal é enviado por um satélite com a posição conhecida o tempo que leva para chegar ao recetor. Este tempo é chamado tempo de propagação e a distância do recetor ao emissor é obtida a partir deste tempo. No entanto, para calcular a localização, é necessário receber sinais de múltiplos satélites, permitindo triangulação entre eles [24].

Para obter uma posição com a mínima margem de erro, o recetor deve receber sinais de pelo menos quatro satélites. Esta necessidade decorre da sincronização entre os relógios do recetor e dos diferentes satélites. O recetor não está sincronizado com o relógio do sistema GNSS, reduzindo assim a complexidade do sistema e mitigando os efeitos da teoria da relatividade. Com quatro satélites, é possível determinar a latitude, longitude, altitude e o desvio de tempo do relógio do recetor em relação ao relógio do sistema [24].

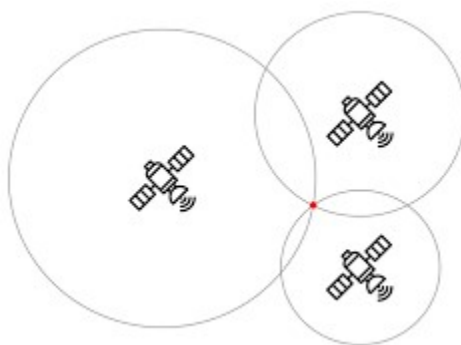


Figura 12 – Funcionamento dos Recetores GNSS [26]

Através da triangulação destas medições, é possível estabelecer a posição do recetor com uma certa precisão e confiabilidade, como ilustrado na Figura 12. A precisão da localização aumenta à medida que aumenta o número de satélites em comunicação com o recetor. Portanto, o desempenho do GNSS depende muito do ambiente em que o recetor está de forma a não cobrir a área onde o recetor se encontra [24]. Para a triangulação apresentada na Figura 12, consideram-se apenas três emissores de sinal, o que é o mais comum. No entanto, para a localização por satélite, é necessário comunicar com quatro emissores, devido à necessidade de sincronização de relógios entre o recetor e os satélites.

2.4.2 Sistemas de Navegação Inerciais

Um sistema de navegação inercial (INS), que é um tipo de sistema de localização relativa, utiliza um conjunto de unidades de medição inercial (IMU) para medir grandezas físicas nos três eixos.

Um IMU é um dispositivo de medição inercial que gera uma solução independente de navegação tridimensional, e geralmente usa uma combinação de acelerómetros, de giroscópios e, às vezes, de magnetómetros. Para medir grandezas físicas nos três eixos, é necessário ter três dispositivos inerciais, cada um medindo em um eixo perpendicular aos outros dois. Observando a Figura 13 o “*Navigation Process*” vai integrar as saídas do IMU dando a posição, a velocidade e a orientação de navegação [24]. Os giroscópios medem a taxa angular que vai ser utilizada pelo processador de navegação para medir a atitude de navegação do INS. Os acelerómetros vão medir as forças da gravidade em diferentes eixos e desta forma conseguir quantificar esta força num vetor de direcção. É usado um modelo de gravidade, que leve em consideração a influência da força gravitacional para obter a aceleração de uma dada força numa dada posição. O valor da velocidade é obtido integrando o valor da aceleração, de forma semelhante a posição é calculada através da integração da velocidade. Os erros numa navegação inercial aumentam à medida que os erros do acelerómetro e do giroscópio são incrementados. Estes erros dependem da qualidade dos sensores utilizados [24].

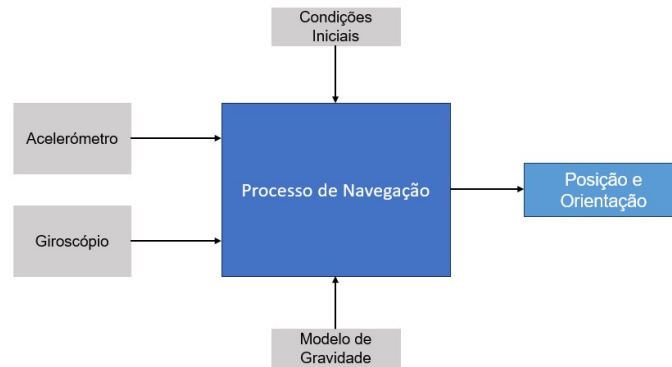


Figura 13 – Processo Navegação INS

2.5 Sensores

Os robôs móveis têm a habilidade de se movimentarem autonomamente em ambientes desconhecidos. No entanto é necessário integrar sistemas sensoriais capazes de recolher informações do ambiente permitindo ao robô reagir de maneira apropriada a eventos inesperados durante a navegação. A qualidade da reação do robô é diretamente proporcional à quantidade e precisão das informações fornecidas pelos sensores. Por esta razão um robô deve ter capacidades sensoriais semelhantes às de um ser humano, recebendo várias informações do mundo exterior por meio de vários sensores de forma a basear suas reações num ou mais sensores específicos.

A robótica móvel emprega uma ampla variedade de sensores, sendo alguns deles usados apenas para medir grandezas como a temperatura interna ou a velocidade de rotação dos motores, outros, mais sofisticados, são usados para adquirir informações do ambiente externo ou para determinar a posição global do robô.

Os sensores utilizados em robôs autónomos podem ser divididos em duas categorias principais: os sensores internos e os sensores externos. Os dispositivos de deteção internos fornecem informações sobre os estados internos do robô, como sua posição, a aceleração, a velocidade e o binário. Por outro lado, os dispositivos de deteção externos são usados para detetar condições externas do robô, de que são exemplo os sistemas de deteção infravermelho e ultrassónico, as câmaras *Red, Green and Blue* (RGB) e os sistemas de alerta de proximidade [27].

Por outro lado, existe uma categoria que se refere à forma como a aquisição da informação é efetuada, o que distingue os sensores de ativos ou passivos. Os sensores passivos apenas recebem informações do ambiente externo, como por exemplo, sensores temperatura/humidade, microfones e câmaras. Já os sensores ativos precisam emitir energia para o ambiente, sendo a resposta do ambiente a essa energia a informação que recebem. Embora estes sensores tenham melhor desempenho, estes são suscetíveis a interferências. Exemplos de sensores ativos são o GPS, os *encoders*, os sensores ultrassónicos e os lasers [28].

2.5.1 Sensores de Orientação

A capacidade de auto-localização de um robô durante a navegação é extremamente importante, sendo que alguns robôs utilizam a sua auto-localização com recurso a sistemas de visão, que por vezes podem não funcionar satisfatoriamente devido à elevada capacidade de processamento exigida por este sistema. Para tal utilizam-se sensores que permitem obter a orientação e localização em relação a um ponto de referência ou sistema de coordenadas.

A) Acelerómetros

Um acelerómetro é um tipo de sensor que mede a aceleração de um dado objeto em movimento, calculando a direção deste. A aceleração é uma grandeza vetorial que descreve a mudança da velocidade de um objeto ao longo do tempo. Ou seja, um acelerómetro é capaz de medir a variação da velocidade do objeto em cada um dos três eixos espaciais: X , Y e Z . Este sensor é um tipo de sistema micro-eletromecânico, *Micro Electro Mechanical Systems* (MEMS), que usa um circuito integrado de silício. Esses sensores convertem o movimento mecânico num sinal elétrico, utilizando componentes piezoelétricos, piezo-resistivos e capacitivos [29].

O princípio de funcionamento de um acelerómetro pode aplicar diferentes conceitos físicos, mas a grandeza medida vai de encontro ao tipo de tecnologia utilizada. O dispositivo pode ser conceptualmente equivalente a um sistema físico, como se pode observar na Figura 14, composto por uma massa ligada a uma mola de frequência natural/ressonante muito superior às frequências de funcionamento do acelerómetro. A segunda lei de Newton afirma quando a massa é sujeita a acelerações, a sua força resultante faz com que a mola se estique, criando uma força elástica em sentido contrário. A aceleração ao longo do eixo do acelerómetro pode ser determinada a partir do comprimento da mola ou da força necessária para contrariar o deslocamento da massa. No entanto, como qualquer massa está sempre sujeita à atração gravitacional, os acelerómetros registam uma aceleração fictícia oposta à gravidade.

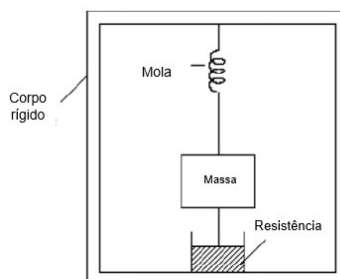


Figura 14 – Funcionamento Acelerómetro [30]

B) Giroscópios

Os giroscópios são dispositivos que têm como função medir e manter a orientação de um objeto em relação a um ponto de referência. O princípio de funcionamento baseia-se na conservação do momento angular, em que quando um rotor gira num eixo a sua rotação produz binário numa ou mais direções. Quando o giroscópio é sujeito a uma força ou binário externo, a rotação do rotor produz uma força de reação que é perpendicular à força ou binário aplicado, mantendo assim a orientação do giroscópio [29].

C) Magnetómetros

Os magnetómetros são sensores usados em robótica para medir a intensidade e direção do campo magnético local, fornecendo informações cruciais sobre a orientação do robô em relação ao campo magnético terrestre ou outros objetos.

Os magnetómetros baseiam-se no magnetismo em materiais ferromagnéticos ou em correntes elétricas, estas dividem-se em duas tecnologias sendo os magnetómetros de efeito Hall e os magnetómetros baseados em fluxo [29].

Os magnetómetros de efeito Hall, como se pode observar na Figura 15, são um tipo de sensor que mede campos magnéticos utilizando o efeito de Hall. O sensor tem uma placa de semicondutor com uma corrente elétrica a fluir através desta, e quando este semicondutor é exposto a um campo magnético externo, é aplicada a força de Lorentz resultando assim no movimento de eletrões para um lado da placa, gerando assim uma diferença de potencial denominada por tensão de Hall. Esta tensão é diretamente proporcional à intensidade do campo magnético aplicado, e o sensor mede a tensão de Hall e converte esta numa leitura de campo magnético.

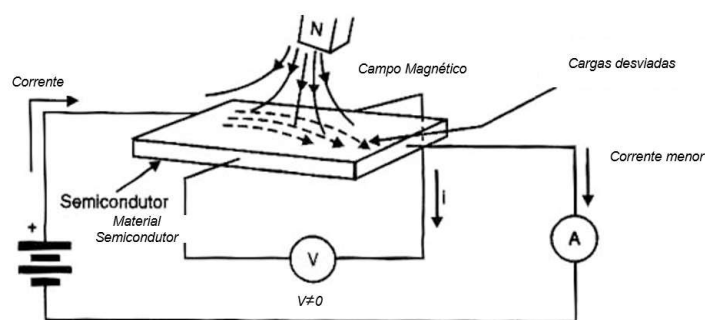


Figura 15 – Magnetómetro efeito de Hall [31]

Já os magnetómetros baseados em fluxo (fluxgate), como se pode observar na Figura 16, medem o campo magnético detetando as mudanças no fluxo magnético através das bobinas enroladas em torno do material ferromagnético. A mudança no fluxo magnético é convertida num sinal elétrico que pode ser lido pelo robô [29].

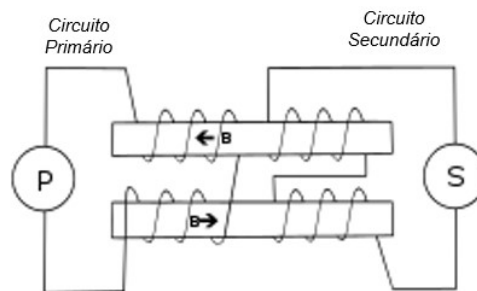


Figura 16 – Magnetômetro fluxgate [32]

Apesar de projetados para detetar o campo magnético da Terra, os compassos magnéticos podem ser influenciados por perturbações do meio envolvente. Tais perturbações podem surgir de campos magnéticos provenientes de outros dispositivos e da presença de materiais com permeabilidade magnética diversa. Um modelo elementar para tais distorções envolve a definição de uma escala e a polarização, as quais, devido à sua forte relação com o ambiente, devem ser avaliadas e corrigidas sob condições operacionais semelhantes [29].

D) *Inertial Measurement Unity (IMU)*

Um IMU é um dispositivo que integra um giroscópio, um acelerómetro e por vezes um magnetómetro, sendo que cada um destes sensores integra 3 eixos espaciais, proporcionando assim um IMU com 9 graus de liberdade. Normalmente um IMU, Figura 17, costuma ter este conjunto de sensores unidos para a fusão de dados sensoriais de modo a fornecer medidas de orientação e direção mais fidedigna.



Figura 17 – Exemplo de um IMU

2.5.2 Encoders

Estes tipos de dispositivos são utilizados para determinar as condições internas e a cinemática de um robô, permitindo o controlo da velocidade angular e da posição das rodas ou outras articulações movidas por motores. Os *encoders* funcionam utilizando um feixe de luz que vai interagir com um disco perfurado, convertendo um movimento linear ou rotativo numa série de impulsos elétricos de onda quadrada. Existem vários tipos de *encoders*, que se distinguem pela sua aplicação e pela sua resolução. Uma vez que estes sensores têm muitas aplicações além da robótica, tem-se verificado uma redução no seu custo e melhorias nas suas capacidades [33].

2.5.3 Sensores Ultrassónicos

Os sensores ultrassónicos como se mostram na Figura 18 são dispositivos que utilizam ondas sonoras de alta frequência para medir a distância entre o sensor e um objeto. Esses sensores irradiam um sinal ultrassónico e medem o intervalo de tempo necessário para que esse sinal volte após ser refletido pelo objeto. A partir desse tempo de retorno, o sensor é capaz de calcular com grande exatidão a distância até o objeto [34].



Figura 18 – Sensores Ultrassónicos

Este tipo de sensores continua a ser um dos mais utilizados em robótica. O seu baixo custo, alta precisão e facilidade com que é determinada a distância a um obstáculo, tornam estes sensores a primeira escolha quando se pretende desenvolver um algoritmo de deteção e desvio de obstáculos. Contudo, estes sensores apresentam também algumas limitações visto que são bastante suscetíveis a interferências, e dependem fortemente das características refletivas dos materiais presentes no meio envolvente. Possuem também um tempo de ciclo relativamente elevado, o que pode, consoante a velocidade do robô ser uma limitação [34].

2.5.4 Sensores LIDAR

O LIDAR refere-se a um tipo de sensor utilizado na robótica e muitas outras aplicações e serve para medir a distância bem como criar mapas num ambiente 2D ou 3D, como se pode observar na Figura 19. O princípio de funcionamento de um LIDAR passa pela emissão de impulsos laser e através destes é possível medir o tempo que este é refletido no sensor. Com base nesse tempo, o sensor pode determinar a distância até ao objeto com uma elevada precisão [35].



Figura 19 – Exemplo de LIDAR existentes no mercado

Este tipo de sensor é usado em muitos sectores, tais como automação, infraestruturas, robótica, UAV/drones, mapeamento e muito mais. Têm uma fonte de luz independente e apresentam um elevado desempenho em diversas condições climáticas e de iluminação [36]. Estes são classificados de acordo com o número de impulsos de luz que podem ser direcionados para uma superfície e o método de varrimento na superfície. Para além disto os sensores LIDAR podem ser 2D ou 3D e na Tabela 2 pode ser verificado a diferença entre estes [35].

Tabela 2 – Comparação LIDAR 2D vs LIDAR 3D

<u>LIDAR 2D</u>	<u>LIDAR 3D</u>
Fornece dados relativamente à distância e posição de um dado objeto em 2 dimensões (horizontal e vertical).	Fornece dados relativamente à distância e posição de um dado objeto em 3 dimensões (horizontal, vertical e profundidade).
Interpretação dos dados é mais simples com menos custos	Interpretação dos dados é mais complexa e mais oneroso
É apropriado para aplicações menos exigentes, como por exemplo a deteção de obstáculos num ambiente simples e plano.	É apropriado para aplicações exigentes com alto nível de detalhe, como por exemplo o mapeamento 3D e a deteção de objetos e a reconstrução destes.
A taxa de amostragem é mais alta do que o LIDAR 3D, o que torna mais eficiente na deteção de objetos em movimentos rápidos.	A taxa de amostragem é mais baixa que num LIDAR 2D, e neste caso limita a deteção de objetos em movimentos rápidos
Baixa capacidade de reconstrução de objetos e ambientes.	Grande capacidade de reconstrução de objetos e do ambiente.
Utilizado em sistemas de navegação autónoma, onde o ambiente é estruturado e simples.	Utilizado em aplicações que requerem maior resolução como o mapeamento 3D e em aplicações em ambientes não estruturados.

Os sensores LIDAR são fundamentais para a navegação autónoma, além de proporcionar visão 3D de alta precisão do ambiente envolvente, fornecem uma nuvem de pontos como se mostra na Figura 20, o que permite realizar a manipulação de dados para a deteção de objetos permitindo que os robôs durante a sua navegação detetem e meçam estes pontos em tempo real, criando um mapa preciso de um dado ambiente em constante mudança permitindo assim uma navegação segura. Devido à sua precisão na medição de distâncias, os sensores LIDAR podem detetar e evitar objetos até 200 metros, mesmo em condições climáticas e de iluminação desafiadoras [36] como se pode observar na Figura 20.

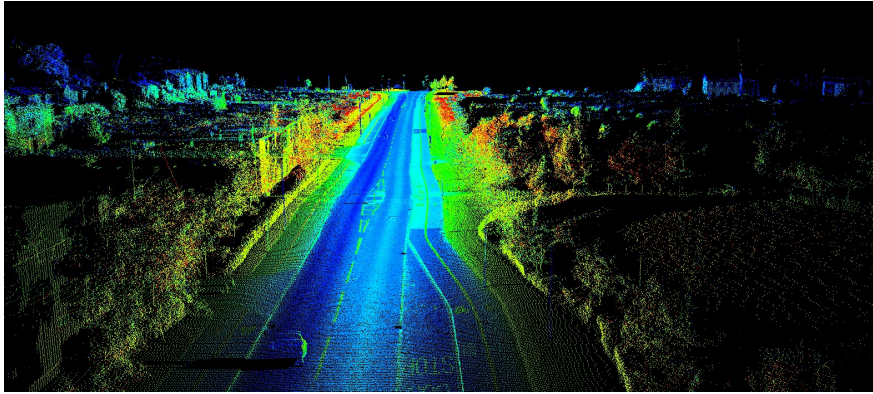


Figura 20 – Exemplo de uma Point Cloud fornecida por um sensor LIDAR

2.5.5 Câmaras

A visão é um dos sentidos mais poderosos do ser humano, devido à quantidade de informação que é absorvida num dado ambiente. Portanto, é compreensível o empenho e a dedicação empregue no desenvolvimento de sensores para robôs capazes de imitar as capacidades visuais humanas. As câmaras são um componente indispensável para os ARMs e com o avanço da inteligência artificial e com a utilização de câmaras permitem que os ARMs sejam capazes de realizar ações semelhantes às humanas e desta forma aperfeiçoar a navegação autónoma para inúmeras aplicações. Uma das principais funções das câmaras é auxiliar na navegação num dado ambiente, ajudando também a perceber que tipo de objetos se encontram presentes, permitindo dessa forma tomar decisões mediante aquilo que se deteta.

Os principais tipos de câmaras usadas nos AMRs são:

- Câmaras RGB utilizam um sistema de cores aditivas em que o Vermelho (*Red*), o Verde (*Green*) e o Azul (*Blue*) são combinados de várias formas. Estas câmaras captam imagens como o ser humano está habituado a ver, mostrando um determinado objeto com cores reais [37];
- Câmaras multiespectral contém múltiplos sensores e cada um destes com um filtro de alta qualidade para conseguir captar diferentes espectros. Este tipo de câmara é muito utilizado em aplicações de agricultura para identificar diferentes índices de vegetação [37];
- Câmaras Depth permitem capturar informações sobre as cores de um ambiente e também calcular a distância entre a câmara e um objeto num dado ambiente. A tecnologia presente neste tipo de dispositivos serve para medir distâncias até um objeto, sendo estas o LIDAR e o *Time-of-Flight* (TOF). A medição da profundidade é baseada na emissão de um sinal (luz ou laser) em direção ao ambiente e desta forma medir o tempo que o sinal demora até ser refletido e com base neste tempo a câmara consegue calcular a distância ao objeto [38];
- Câmaras térmicas consistem num dispositivo que deteta energia infravermelha e a converte numa imagem visual. Esta câmara é composta por

uma lente, um sensor térmico e um processador eletrônico. A lente da câmara foca a energia infravermelha do sensor. As resoluções deste tipo de câmaras são mais baixas em comparação com outro tipo de câmaras, dado que os sensores térmicos necessitam de detetar energia com comprimentos de onda muito maiores que os da luz visível, impondo assim que cada elemento do sensor seja significativamente maior. Este tipo de câmara pode abranger vários tipos de aplicações devido à sua grande precisão ao medir o calor detetado [39].

2.5.6 GPS

Tal como referido na seção 2.4 o GPS (Figura 21) é um GNSS desenvolvido pelo Departamento de Defesa dos Estados Unidos da América, que teve como objetivo fortalecer e contornar as limitações existentes nos sistemas de localização.



Figura 21 – Exemplo de um sensor GPS

Os sistemas GNSS, incluindo o GPS, possuem limitações tecnológicas que levam a erros nas medições. A principal limitação é a necessidade de comunicação constante com pelo menos quatro satélites para calcular a posição do recetor, o que torna o seu uso em espaços fechados inviável porque pode resultar na perda de visibilidade de alguns satélites e, consequentemente, na perda de dados de localização [24].

Os erros nas medições do GPS podem ser divididos em duas categorias: erros de modo comum e erros de modo não comum. Erros de modo comum são erros que afetam todos os recetores de GPS numa ampla área (50-200 km) e ocorrem devido a atrasos na transmissão de sinal pela atmosfera terrestre e erros de relógio do satélite. Já os erros de modo não comum são específicos do local e do recetor e resultam de propagação multipercurso de sinais de rádio e do ruído do recetor. Embora os satélites GNSS tenham relógios atómicos altamente precisos, é praticamente impossível evitar pequenos erros de tempo, sendo estes relativamente pequenos numa escala milissegundos. Esse tipo de erro é conhecido como erro de relógio e pode ser atribuído a fatores relativísticos e desvios de órbita dos satélites [24].

Os erros de modo não-comum são específicos do recetor utilizado e das condições do ambiente em que é utilizado. Com as melhorias nos sistemas de GNSS, esses erros podem ser a maior fonte de imprecisão nas medições. O erro de multipercurso ocorre quando o sinal GPS é refletido nas proximidades do recetor, causando a receção de múltiplos sinais com ligeiros atrasos em relação ao sinal direto. Esse fenómeno pode afetar a correlação entre o sinal recebido e a referência criada no recetor, levando a medições imprecisas, cuja amplitude aumenta com o atraso entre o sinal original e o sinal refletido. Embora existam métodos para reduzir o impacto desse erro, a melhor estratégia é evitar utilizar o recetor em locais onde a probabilidade de reflexões é elevada. Além do problema da reflexão, há também a possibilidade de ocorrer *shadowing*, em que o sinal direto do GPS é enfraquecido por um objeto parcialmente bloqueador, como uma árvore, e o sinal refletido detetado é mais forte [24]. Pode-se observar este tipo de erros na Figura 22 onde mostra a ocorrência quer de multipercurso quer de *shadowing*.

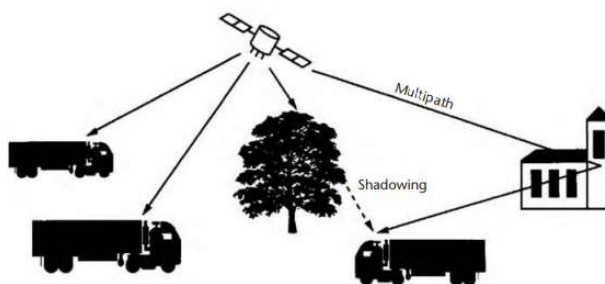


Figura 22 – Erros de Multipercurso e Shadowing

2.6 Sistemas de Coordenadas GPS

Um sistema de coordenadas GPS pode ser representado por vários sistemas de coordenadas, sendo o sistema Latitude e Longitude e o sistema UTM, os mais utilizados [29].

2.6.1 Latitude e Longitude

As coordenadas geográficas de latitude e longitude são fundamentais para a localização e movimentação no espaço terrestre, sendo apresentadas em graus. Embora complementares, essas medidas possuem diferenças significativas entre si. A latitude é a medida em graus da distância de um ponto da superfície terrestre em relação à Linha do Equador, variando de 0° a 90° em direção ao norte ou sul. Já a longitude é a medida em graus da distância de um ponto da superfície terrestre em relação ao Meridiano de Greenwich, variando de 0° a 180° em direção leste ou oeste [24] conforme se mostra na Figura 23

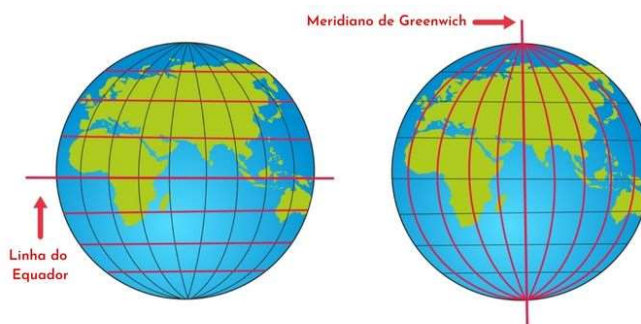


Figura 23 – Localização do Equador e do Meridiano de Greenwich

2.6.2 UTM

O sistema de coordenadas UTM ou Coordenadas Universais Transversas de Mercator, é um sistema que tem sido utilizado pelo exército há algum tempo e é utilizado para informar a posição nos sistemas inerciais de navegação na Terra. As coordenadas UTM permitem que a saída do sistema seja facilmente correlacionada com a posição em um mapa plano. Ou seja, como a Terra é quase esférica e em termos matemáticos, é uma “superfície não desenvolvível”, isto significa que qualquer método para a achatar em um plano pode envolver alguma distorção. Embora muitas vezes não seja percebido, a projeção de Mercator, que é usada para criar um mapa plano do mundo, causa distorções enormes na forma da superfície terrestre. Essa projeção é construída projetando a superfície da Terra num cilindro tangente à Terra no equador [29]. A projeção de Mercator usa um cilindro tangente ao longo de um meridiano e as coordenadas UTM nas direções norte e leste são conhecidas como “*northing*” e “*easting*” e geralmente são descritas em metros [30].

2.7 Filtros Digitais

Como referido na seção 2.5, os AMRs são compostos por uma vasta gama de sensores. Dados provenientes de sensores são suscetíveis a ruído provenientes de diversas fontes, tais como interferências ou variações de temperatura. Isso resulta na introdução de incertezas na confiabilidade dos dados adquiridos. Portanto, é essencial implementar filtros para mitigar o ruído, eliminando assim imprecisões nas medições que os sensores realizam.

Existem duas categorias principais de filtros: os analógicos e os digitais, estas duas categorias diferem significativamente na sua estrutura física e na forma como operam [40]. Um filtro analógico faz uso de circuitos eletrônicos analógicos, construídos com resistências, condensadores e amplificadores operacionais, a fim de criar o efeito de filtragem desejado. Estes circuitos são amplamente utilizados em diversas aplicações, incluindo a redução de ruído, melhoria de sinais de vídeo, equalização gráfica, entre outros.

Para projetar o circuito de filtro analógico com base em requisitos específicos, existem técnicas consolidadas e bem estabelecidas. Em cada etapa desse processo, o sinal submetido à filtragem é uma forma de tensão ou corrente elétrica, que é diretamente proporcional à quantidade física que está sendo medida/processada [40]. Um filtro digital opera sobre os sinais que foram convertidos para a forma digital, ou seja, realiza-se um processamento específico nos dados digitais, resultando num novo sinal que contém as modificações em relação ao sinal original. Os filtros digitais têm várias aplicações possíveis, incluindo a redução de ruído, o realce de informações específicas e a separação de informações em diferentes frequências. Estes são utilizados para ajustar ou melhorar características específicas de sinais digitais, tornando-os mais adequados para análise, transmissão ou armazenamento. Neste trabalho é feito um estudo sobre os filtros digitais [40].

2.7.1 Vantagens dos Filtros Digitais

Os filtros digitais apresentam inúmeras vantagens quando comparados com os filtros analógicos, podem ser programáveis e o seu funcionamento pode ser armazenado em memória, eliminando a necessidade conceber um circuito eletrónico de filtragem sempre que exista uma alteração.

Além disso, os filtros digitais não requerem a construção de circuitos personalizados para cada aplicação específica, ao contrário dos filtros analógicos. Uma vantagem adicional dos filtros digitais é que eles não são afetados por variações de temperatura, ao contrário dos filtros analógicos que podem ser sensíveis a estas variações. Os filtros digitais são altamente versáteis, permitindo o processamento de sinais de diversas maneiras e a capacidade de adaptação do filtro de acordo com as características do sinal de entrada [40].

2.7.2 Classificação de Filtros

O uso de um filtro digital pode ser dividido em três categorias: domínio do tempo, domínio da frequência e personalizado. Os filtros no domínio do tempo são usados quando os detalhes da informação são incorporados na configuração da forma de onda de um sinal específico. A filtragem no domínio do tempo é usada para suavização do sinal, eliminação da componente contínua, ajuste da forma de onda, entre outras. Por outro lado, os filtros no domínio da frequência são utilizados quando as características importantes da informação estão contidas na amplitude, frequência e fase das componentes sinusoidais. A finalidade principal destes filtros é separar um conjunto de frequências de outras, destacando ou isolando partes específicas do espectro das frequências. Na Figura 24 mostra-se a diferença de forma gráfica entre o domínio do tempo e domínio da frequência [40].

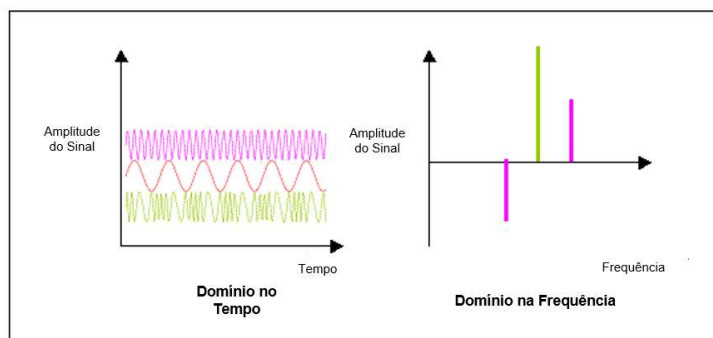


Figura 24 – Domínio do tempo vs Domínio de frequência [41]

Para uma ação mais especializada do filtro, diferente das quatro respostas básicas de um filtro: passa-alto, passa-baixo, passa-banda e rejeita-banda, são necessários filtros personalizados. Estes filtros são concebidos para atender requisitos específicos e podem ser mais complexos e adaptados às necessidades precisas da aplicação em questão [40]. Os filtros digitais podem ser classificados como *Finite Impulse Response* (FIR) e *Infinite Impulse Response* (IIR). Uma das distinções entre esses dois tipos de filtros está relacionada com o facto de um filtro FIR ter uma resposta ao impulso de duração finita enquanto um filtro IIR poder produzir uma resposta ao impulso de duração infinita [40].

A) Filtros IIR

Um filtro IIR gera uma resposta infinita ao impulso de um sistema dinâmico. Devido ao mecanismo de realimentação inerente contido nesses filtros, o filtro utiliza uma porção do sinal de saída (já filtrado) reintroduzindo-o novamente no processo de filtragem. Os Filtros IIR são utilizados em sistemas que precisam de processar sinais contínuos ao longo do tempo, isto significa que a saída do filtro depende das entradas atuais e de estados anteriores, o que possibilita uma resposta dinâmica e adaptada aos sinais de entrada. O diagrama de blocos descrito na Figura 25 representa a estrutura de um filtro IIR de segunda ordem:

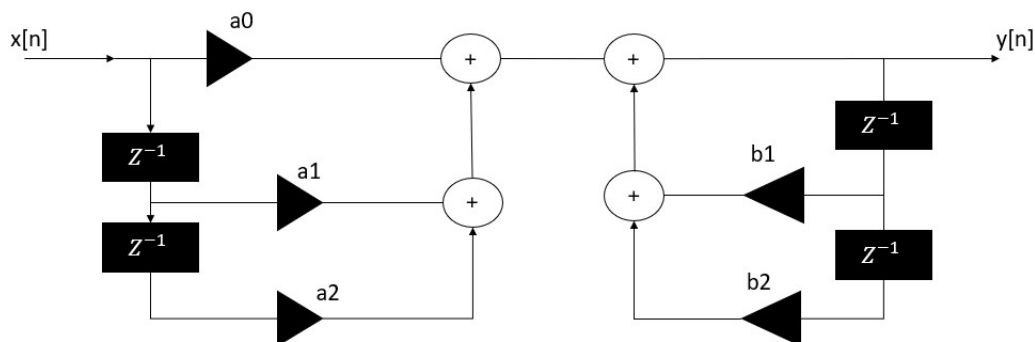


Figura 25 – Exemplo de uma estrutura de um filtro IIR

Observando a Figura 25, o filtro IIR opera de tal forma que considerada as entradas presentes, as entradas passadas e também o valor das saídas passadas. A estrutura básica de um filtro IIR é composto por duas partes sendo estas: a parte "*feedforward*" (alimentação direta) e a parte de "*feedback*" (realimentação). Na parte *feedforward* os coeficientes (a_0, a_1, a_2) multiplicam com as entradas de sinal ($x[n]$). Na parte de *feedback* os coeficientes (b_1, b_2) multiplicam com as saídas passadas ($y[n]$) e resulta numa realimentação que origina uma resposta do filtro de impulso infinita. A equação geral de um filtro IIR pode ser expressa da seguinte forma [42]:

$$y(n) = \sum_{i=1}^N b_i y(n-i) + \sum_{j=0}^N a_j x(n-j) \quad (16)$$

onde:

- $y[n]$ é a saída do filtro no instante n ;
- $x[n]$ é a entrada do filtro no instante n ;
- a_j são os coeficientes do filtro que determinam a resposta ao impulso para as entradas;
- b_i são os coeficientes do filtro que determinam a resposta ao impulso para as saídas.

Na equação acima é notório que a saída ($y[n]$) é a soma ponderada das entradas e saídas passadas. A equação 16 pode ser reescrita usando a transformada de Z:

$$H(Z) = \frac{Y(Z)}{X(Z)} = \frac{\sum_{k=0}^N a_k Z^{-k}}{1 - \sum_{k=1}^N b_k Z^{-k}} \quad (17)$$

A ordem do filtro é normalmente determinada pela relação entre os coeficientes de alimentação direta (N_x) e os coeficientes de realimentação (N_y). O grau de semelhança da resposta em frequência de um filtro IIR a um filtro ideal está relacionada com a ordem do filtro, ou seja, quanto maior for a ordem de um filtro mais este se aproxima de um comportamento ideal. Isso é observado na figura 26 onde mostra a resposta de frequência de filtros Butterworth passa-baixo de diferentes ordens. Quanto maior a ordem do filtro, mais íngreme é a queda no ganho do filtro [42].

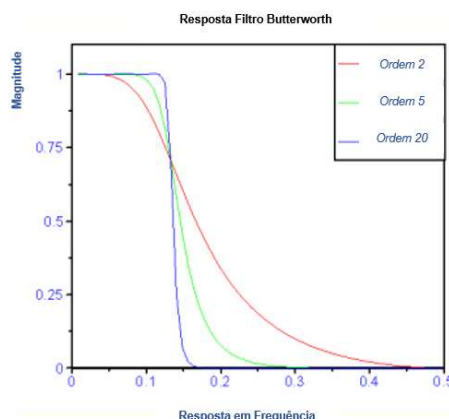


Figura 26 – Gráfico que mostra o comportamento de ordens dadas a um filtro IIR [42]

A estabilidade dos filtros IIR depende de diversos fatores sendo estes relacionados com o sistema, o que pode resultar que a saída cresça indefinidamente como a resposta a uma entrada de valor finito. Visto que o filtro pode atingir ordens mais baixas, torna o filtro mais eficiente em termos computacionais, no entanto a implementação deste tipo de filtro é mais complexa [42].

B) Filtros FIR

Um filtro FIR gera uma resposta ao impulso de duração finita. Isto significa que a resposta do filtro responde apenas a uma entrada, num dado número finito de amostras, e após este período a resposta fixa em zero durante um determinado tempo. Em outras palavras um filtro do tipo FIR cria uma saída com base numa combinação linear de amostras da entrada atual e entradas passadas, onde os coeficientes do filtro são fixos e não dependem das saídas anteriores do filtro, ou seja, a quantidade de amostras para processamento corresponde a um número finito de valores de entrada. Na Figura 27 apresenta-se o diagrama de blocos da estrutura de um filtro FIR [42]:

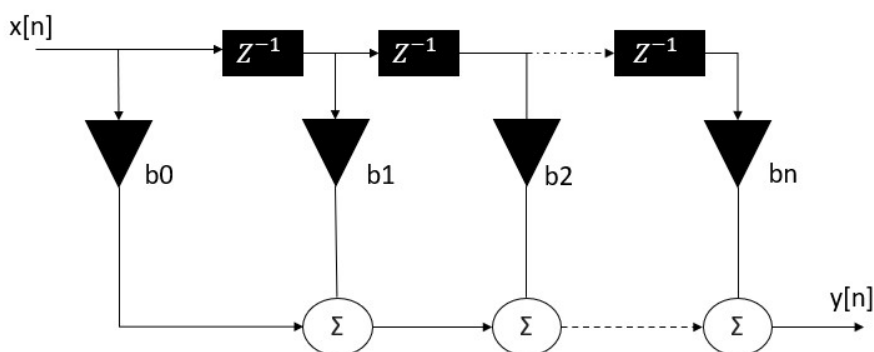


Figura 27 – Exemplo de uma estrutura de um filtro IIR

A equação geral de um filtro FIR pode ser expressa da seguinte forma [42]:

$$y(n) = \sum_{k=0}^{N-1} b_k \times x_{n-k} \quad (18)$$

onde:

- $y[n]$ é a saída na amostra n .
- $x[n]$ é a entrada na amostra n .
- $b[k]$ são os coeficientes do filtro, onde k varia de 0 a $N - 1$.
- N é a ordem do filtro, que determina quantos coeficientes são usados na equação.

A equação geral mostra que a saída é uma soma ponderada das amostras de entrada atuais e passadas, descrevendo os coeficientes $h[k]$ o peso de cada amostra de entrada. É possível reescrever a equação 18, usando a transformada de Z:

$$H(z) = h[0] + h[1] \times z^{-1} + h[2] \times z^{-2} + \dots + h[M] \times z^{-M} \quad (19)$$

onde:

- $H(z)$ é a função de transferência.
- $h[k]$ são os coeficientes do filtro.
- $z^{-1}, z^{-2}, \dots$ são termos da transformada Z que representam atrasos de amostragem.

A principal característica de um filtro FIR é a sua estabilidade, dado que a resposta ao impulso termina após um número finito de amostras e desta forma não existe realimentação envolvida. Apesar deste tipo de filtros ser mais fácil de projetar em comparação com os IIR, a aplicação deste tipo de filtros requer filtros de ordens mais elevadas e consequentemente maior esforço de processamento [42].

2.7.3 Tipos de Filtros

Os filtros digitais podem ser categorizados em diversos tipos, no entanto, neste tópico vão ser abordados quatro tipos principais de filtros IIR sendo estes: passa-baixo, passa-alto, passa-banda e rejeição de banda e o filtro FIR de média móvel. Cada filtro tem uma aplicação específica no processamento digital de sinal, e a escolha deste vai depender dos requisitos do sistema, recursos computacionais e característica do sinal.

A) Filtros Passa-Baixo

Os filtros passa-baixo são os mais comuns. A Figura 28 ilustra o funcionamento deste tipo de filtro, o qual permite a passagem de baixas frequência e atenua as altas frequências. Na Figura 28 Ω_p e Ω_s representam as frequências de corte da banda passa-banda e da banda de rejeição, respetivamente. A banda de passagem

corresponde às frequências do sinal de entrada que passam para a saída sem atenuação. A banda de rejeição corresponde à gama de frequências do sinal de entrada que são rejeitadas pelo filtro e a banda de transição corresponde à zona intermédia entre as duas bandas na qual a atenuação varia linearmente [43].

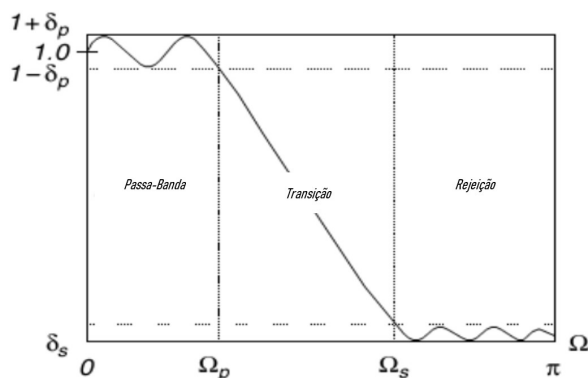


Figura 28 – Resposta frequência de um filtro Passa-Baixo [43]

B) Filtros Passa-Alto

O funcionamento do filtro passa-alto, como a Figura 29 mostra, é o inverso do filtro passa-baixo, onde permite a passagem de frequências altas e atenua a amplitude das frequências inferiores à frequência de corte [43].

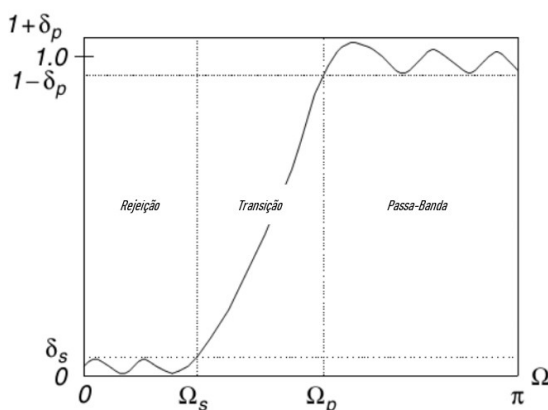


Figura 29 – Resposta frequência de um filtro Passa-Alto [43]

C) Filtro Passa-Banda

Este tipo de filtro é mais comum em sistemas de comunicação, e conforme ilustra a Figura 30, este permite a passagem de frequências numa determinada banda, atenuando altas e baixas frequências. Na Figura 30 Ω_{pL} e Ω_{sL} são os limites inferiores e superiores da passa-banda e Ω_{pH} e Ω_{sH} são a frequência limite superior da banda passa-banda e o limite inferior da frequência da banda de rejeição de banda, respetivamente. Δp é o parâmetro que define a ondulação da banda de passagem [43].

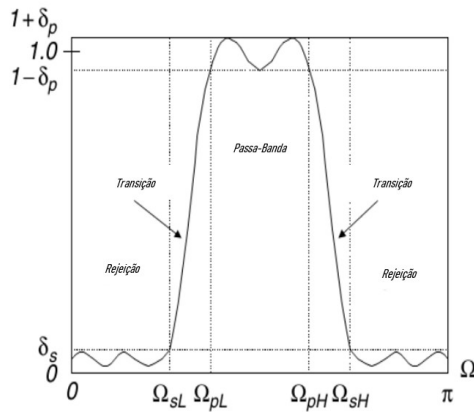


Figura 30 – Resposta em frequência de um filtro passa-banda [43]

D) Filtro Rejeita banda

O filtro rejeita banda conforme mostra a Figura 31, complementa o filtro passa-banda onde se garante um valor de atenuação dentro da banda de rejeição. Desta forma atenua as frequências dentro desta banda, permitindo que sinais com componentes às baixas e altas frequências passem inalterados [43].

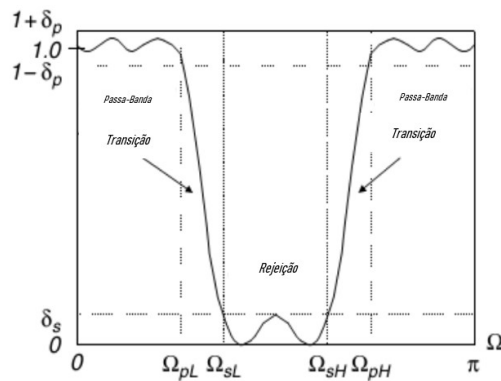


Figura 31 – Resposta em frequência de um filtro rejeição de banda [43]

A) Filtro de Média Móvel

O filtro de média móvel é um filtro FIR, utilizado para suavizar um sinal, eliminando picos e ruído preservando assim o valor médio de um sinal. Este tipo de filtro pode ser obtido através de quatro tipos de metodologias conforme se mostra na Figura 32 [44]:

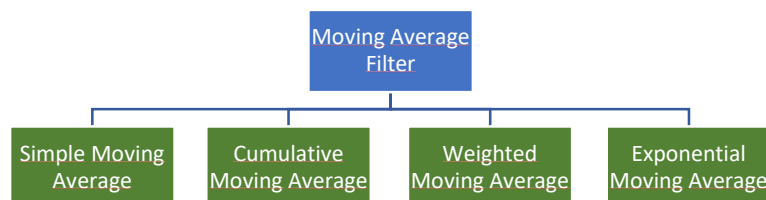


Figura 32 – Tipos de filtros Moving Average

A Média Móvel Simples, ou *Simple Moving Average* (SMA) consiste num filtro simples de média móvel, onde todos os valores têm o mesmo peso com um tamanho fixo. A Média Móvel Cumulativa ou *Cumulative Moving Average* (CMA), é uma variante do SMA que calcula a média cumulativa de um determinado tamanho de amostras até um dado momento, em vez de usar um tamanho fixo como o SMA. Cada valor do SMA é calculado como a média de todos os valores obtidos até um determinado momento. A Média Móvel Ponderada, ou *Weighted Moving Average* (WMA), atribui pesos diferentes as amostras guardadas. Por fim Média Móvel Exponencial, ou *Exponential Moving Average* (EMA), é uma abordagem mais avançada que dá mais peso aos valores mais recentes e diminui o peso dos valores mais antigos [44].

2.8 Fusão Sensorial

Como foi referido a maior parte dos AMRs depende de múltiplos sensores para realizar tarefas de perceção, localização e orientação. Desta forma permite-se que o robô aproveite as diferentes funcionalidades e a capacidade de cada sensor de forma a obter uma capacidade combinada e mais completa em comparação com uma entrada. Por exemplo um AMR pode utilizar um LIDAR ou um sensor ultrassom para medir uma determinada distância, e pode haver casos em que um funciona melhor que o outro. Outro exemplo é robô utilizar um sensor GNSS ou *encoders* para estimar a sua posição. Embora cada sensor forneça dados para o mesmo objetivo, a aquisição de dados pode ser bastante diferente o que resulta em alguns sensores gerarem mais incerteza. Na maioria dos casos este problema está associado à qualidade de cada sensor, ao ruído deste e ao ambiente pelo qual o sensor está exposto. Assim sendo, um sistema de fusão sensorial deve de ser capaz de correlacionar as incertezas e fornecer dados com confiabilidade e robustez.

A fusão sensorial refere-se assim a uma metodologia computacional que combina as medições de vários sensores para que, em conjunto, estes forneçam medições

sobre o sistema medido permitindo melhores resultados que os obtidos da medição de apenas de um sensor. Na área da robótica móvel a fusão de sensores tem sido usada em diversas aplicações de navegação, localização e controlo [45]. Diferentes algoritmos são usados para diferentes níveis de fusão devido às diversas naturezas do processo de fusão. Normalmente, esses incluem inteligência artificial, técnicas de categorização e teoria da probabilidade. Neste tópico irão ser abordados dois algoritmos de fusão sensorial, sendo estes o Filtro de Kalman e Filtro Complementar que vão de encontro às necessidades práticas do trabalho deste projeto.

2.8.1 Filtro de Kalman

O filtro de Kalman é uma técnica que integra dados de medições sensoriais, incluindo ruído, com um modelo dinâmico para aprimorar as correções das estimativas de um sistema. Esse filtro é vulgarmente utilizado na estimativa de posição em sistemas de condução autónoma [46] e regra geral segue um processo de duas etapas: predição e atualização, conforme explicado em [46, 47]. Na etapa de predição, o filtro de Kalman utiliza o modelo dinâmico do sistema para estimar a posição futura do objeto, com base em suas posições e medidas anteriores. Essa estimativa leva em consideração as incertezas e variações associadas ao movimento do objeto e pode ser definida da seguinte forma [46, 47]:

$$X_{k+1} = \varphi_k X_k + W_k \quad (20)$$

onde:

- X_k é o vetor do estado de processo no instante t_k ;
- φ_k é a matriz de transição;
- W_k é o vetor de ruído do processo com uma covariância conhecida.

Na fase de atualização, o filtro integra a previsão anterior com a medida mais recente do objeto, ajustando a previsão com base nas discrepâncias entre a medida real e a estimativa. O filtro leva em consideração as incertezas associadas tanto à medida quanto à previsão, calculando um fator de ponderação (ou ganho) que é aplicado para combinar de forma ponderada as informações da medida e da previsão. Esse fator de ponderação é crucial para obter estimativas mais precisas, dando maior importância às medidas confiáveis e menor importância às medidas menos confiáveis [46]. A equação que descreve esse processo é definida em [46,47]:

$$Z_k = H_k X_k + V_k \quad (21)$$

onde:

- Z_k é o vetor de medição no instante t_k ;
- H_k é a matriz que conecta as medições;
- V_k é o vetor de ruído das medições com uma covariância conhecida.

2.8.2 Filtro Complementar

Durante a navegação autônoma a estimativa da orientação de um robô, em relação a um referencial inercial, é extremamente necessária. Dados provenientes de um IMU podem ser afetados por níveis elevados de ruído, e desta forma um algoritmo de fusão de sensores deve ser usado para processar os dados e obter uma estimativa suave da orientação, mantendo assim um baixo esforço computacional.

O filtro complementar implementa uma abordagem que combina os dados do acelerômetro com os dados do giroscópio a fim de estimar a orientação. Este algoritmo de fusão tem a particularidade de ter uma baixa carga computacional, robustez e uma rápida convergência. O filtro complementar é uma alternativa ao filtro de Kalman Estendido, ou *Extended Kalman Filter* (EKF) devido à sua simplicidade e eficácia, onde faz uma análise no domínio da frequência para filtrar os sinais, permitindo alcançar maior precisão nas estimativas de ângulos de giroscópios a curto prazo e estabilidade nas estimativas de ângulos de acelerômetros a longo prazo. Para tal, este filtro utiliza uma abordagem no domínio da frequência onde é aplicado um filtro passa-baixo às medições do acelerômetro e um filtro passa-alto às medições do giroscópio [48]. A aplicação resulta numa maior estabilidade das leituras do giroscópio durante curtos períodos e numa estabilidade das leituras do acelerômetro durante longos períodos. O diagrama de blocos apresentado na Figura 33 mostra o funcionamento do filtro complementar onde são aplicados ambos os filtros mencionados e na saída é obtido o ângulo final estimado [48].

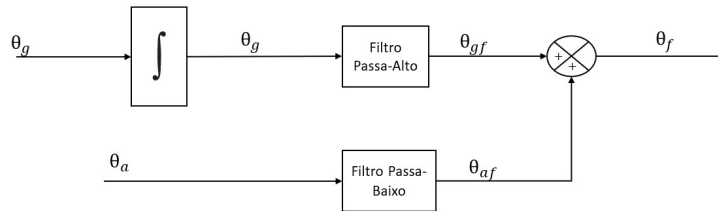


Figura 33 – Funcionamento do Complementary Filter

A essência do filtro complementar é descrita na equação 22, onde $\theta[n]$ corresponde ao ângulo estimado (*roll* ou *pitch*) dado pelo filtro complementar que é igual à estimativa do acelerômetro no instante $[n-1]$ multiplicado uma constante alfa (α), que está entre 0 e 1, e de seguida estes valores são somados à estimativa dada pelos valores do giroscópio onde soma o valor angular no instante anterior, $\theta[n-1]$, com o somatório dos dados fornecidos pelo giroscópio no instante anterior. Estes valores são juntos onde o parâmetro alfa define a taxa de confiabilidade entre os dados do giroscópio e do acelerômetro.

$$\theta[n] = \theta_{acc_{n-1}} \times \alpha + (1 - \alpha) \times \left[\theta_{n-1} + \int \theta_{gyr_{n-1}} \right] \quad (22)$$

onde:

- $\theta[n]$ corresponde ao valor angular atual dado pelo filtro complementar;
- $\theta_{acc_{n-1}}$ corresponde ao valor angular dado pelo acelerómetro no instante anterior;
- α corresponde a uma constante de confiabilidade entre os dados do acelerómetro ou do giroscópio;
- θ_{n-1} corresponde ao valor angular dado pelo filtro complementar no instante anterior;
- $\theta_{gyr_{n-1}}$ corresponde ao valor angular dado pelo giroscópio no instante anterior.

O valor α corresponde a uma constante compreendida entre 0 e 1, permitindo inferir o quanto se confia nos dados, do acelerómetro ou do giroscópio. Então observando a equação 22 se α estiver próximo de 0, significa que é preferível usar os dados provenientes do somatório do giroscópio e caso α estiver próximo de 1 significa que se dá mais relevância aos dados do acelerómetro.

2.9 Visão Computacional

A visão computacional é uma área que se baseia na mimetização do processo visual humano usando *software* e *hardware*. Ela envolve a aquisição, transmissão, processamento e compreensão de informações visuais a partir de imagens ou sequências de imagens. Embora a visão seja uma habilidade comum para pessoas, os princípios por detrás do sistema visual são complexos. Os trabalhos de investigação realizados em visão computacional requerem avanços contínuos, na qual a teoria da computação e a inteligência artificial desempenham um papel relevante.

Os princípios subjacentes à visão computacional compreendem a substituição do cérebro humano por um computador e a substituição do olho humano por uma câmara, de forma que o computador adquira aptidões robustas de reconhecimento por meio de métodos técnicos especializados. Desta forma os seres humanos poderão ser substituídos em tarefas, como sejam, a criação de produtos, a supervisão ou outras atividades, apresentando capacidades semelhantes aos princípios do ser humano. O desenvolvimento da área da visão computacional tem-se tornado muito importante, permitindo que um computador realize um processamento tridimensional através de imagens bidimensionais. A visão computacional desenvolve-se principalmente por meio da inovação e avanços em tecnologias relacionadas tais como a análise estatística de probabilidade e processamento de imagens [49].

Neste trabalho vão ser abordados os princípios básicos do funcionamento da visão computacional, seguido das principais tarefas em que visão computacional está inserida. De seguida, abordam-se os algoritmos existentes para deteção de objetos, finalizando pela explicação do funcionamento do algoritmo YOLOv5 que foi utilizado na metodologia deste trabalho.

2.9.1 Princípio Básico de Funcionamento

Uma utilização eficiente da visão computacional pressupõe o cumprimento de certos requisitos, a ter conta, tais como uma boa iluminação e brilho apropriado no ambiente que pretendemos trabalhar. Assim, são utilizadas câmaras com uma boa definição que permitem recolher imagens e de seguida processá-las. É comum fazer o pré-processamento de imagens como por exemplo a sua segmentação que dará origem a informações adicionais permitindo extrair informação valiosa e crítica para a sua classificação cumprindo o objetivo de se obter a saída desejada de um sistema [49]. Na Figura 34 apresenta-se um diagrama de blocos que mostra de forma mais perceptível, o funcionamento básico da visão computacional.

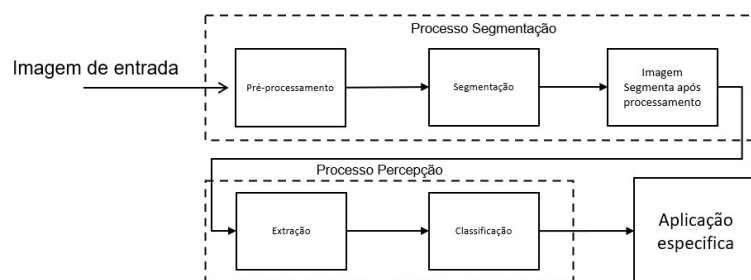


Figura 34 – Funcionamento da visão computacional [49]

2.9.2 Principais tarefas na Visão Computacional

As tarefas em visão computacional têm por finalidade fazer com que os computadores compreendam imagens digitais e procedam à extração, processamento e análise de informações dadas por uma entrada ou mais, com o objetivo de mudar o estado de um sistema. Assim, as tarefas principais que a visão computacional introduz são [50]:

- Classificação de imagens que tem sido uma das áreas mais estudadas por investigadores, e consiste em classificar um conjunto de classes predefinidas usando apenas um conjunto de amostras de imagens que foram previamente classificadas;
- Detecção de objetos, que se refere à percepção e localização de objetos utilizando caixas delimitadoras. A detecção de objetos procura detalhes específicos de classes numa imagem e identifica-os sempre que surjam;
- Segmentação de imagem que se refere à divisão de uma imagem em subpartes ou subobjectos a fim de demonstrar a distinção de vários objetos na mesma imagem;

Para este projeto a tarefa que vai de encontro ao trabalho realizado é a detecção de objetos. Para tal é utilizado o algoritmo de detecção YOLOv5 que vai ser abordado no tópico seguinte.

2.9.3 YOLOv5

O YOLOv5 é um algoritmo de deteção de objetos e é um modelo de visão designado por "*You Only Look Once*" (YOLO). Este algoritmo pertence à família YOLO devido à sua eficiência e velocidade na deteção de objetos em tempo real. O YOLOv5 representa a quinta geração desta série, a qual oferece melhorias significativas em relação às versões anteriores [51]. A sua preferência na comunidade científica sobressaiu devido ao seu excelente desempenho em contexto onde os dados são complicados e ruidosos e à sua simplicidade de usar um conjunto de linguagens populares tais como o Python. A arquitetura do YOLOv5 é composta por 5 secções como se pode observar na Figura 35. O bloco de Entrada faz uma divisão da imagem em secções e cada secção é responsável por detetar uma região específica da imagem, o *Backbone* (espinha dorsal) vai extrair as características das imagens, o *Neck* (pescoço) é a parte intermediária desta arquitetura e realiza uma fusão a fim de melhorar a deteção de um dado objeto, a Deteção vai passar a atribuir uma determinada classe aos objetos detetados e por fim a Saída gera os resultados de deteção como a classe, confiabilidade, localização e tamanho [52].

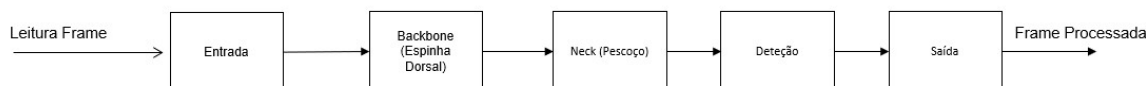


Figura 35 – Estrutura do Algoritmo YOLOv5. (Baseado em [52])

Este sistema utiliza uma rede neural convolucional para recolher e organizar as características de imagens em vários níveis e de seguida une essas características e prepara-as para a etapa de previsão [52]. O YOLOv5 opera em duas fases distintas: treino e inferência. Durante o treino, o modelo é exposto a um extenso conjunto de imagens contendo os objetos que se desejam detetar. Na fase de inferência, o modelo previamente treinado com novas imagens para realizar a deteção dos objetos previamente apreendidos. O YOLOv5 processa a imagem numa única passagem por meio de várias camadas convolucionais, o que resulta numa deteção rápida e altamente confiável [51].

2.10 Sistema Operativo ROS

O *Robotic Operating System* (ROS), é uma estrutura de código aberto que permite aos robôs executarem tarefas. O ROS tem a finalidade de servir como uma plataforma de *software* comum para as pessoas que estão a construir ou a utilizar um robô. O ROS é constituído por várias partes [53]:

1. Um conjunto de *drivers* que permitem ler dados provenientes de sensores e enviar comandos para atuadores, num formato definido;

2. Uma vasta gama de algoritmos fundamentais para a robótica que permitem contruir mapas, navegação, representar e interpretar dados de sensores, planejar movimentos, etc.;
3. Um amplo conjunto de ferramentas que facilitam a visualização do estado de um certo robô e dos algoritmos aplicados, a análise ao comportamento quando ocorrem falhas e o registo de dados sensoriais.

As principais características do ROS são [53]:

- *Peer to Peer* é um sistema no ROS composto por diversos pequenos programas que se interligam e trocam informações por meio de mensagens;
- *Tools-based* envolve uma variedade de ferramentas que têm a capacidade de serem criadas, modificadas e compartilhadas;
- Multilinguagem oferece suporte para a programação em várias linguagens diferentes. Dispõe de uma variedade de ambientes, módulos e bibliotecas adaptadas para diversas linguagens, incluindo C++, Python, LISP, Java, JavaScript, MATLAB, Ruby, Haskell, R, Julia, entre outras;
- Compacto e fomenta a reutilização de código, incentivando os colaboradores a desenvolverem bibliotecas que podem ser compartilhadas e aproveitadas em vários projetos.

O princípio de funcionamento do ROS é baseado numa dinâmica de *nós*. Estes são um conjunto de arquivos executáveis dentro de uma ROS *package*, que permitem comunicar através de tópicos. Os *nós* tanto podem publicar dados como subscrever num tópico.

A publicação de dados ocorre por parte de um *Publisher* que envia uma mensagem para um *nó*, e o *Subscriber* são aqueles que vão ler o conteúdo da mensagem publicada. Há que salientar que na *framework* ROS, dentro de um tópico pode haver apenas um *Publisher* mais infinitos *Subscribers* [53]. No ROS existe uma área de trabalho onde estão inseridas as ROS *packages* que se encontram dentro de uma pasta designada por *src*. Dentro dos *packages* estão os *nós* que vão realizar a publicação ou subscrição de um determinado tópico que está associado a um *nó*. Em cada *package* é obrigatório incluir dois ficheiros sendo estes designados por *CmakeLists.txt*, que serve para compilar o código incluindo num package assim como a gestão das dependências, e outro arquivo essencial é o *Package.xml* que define as propriedades de uma ROS *package* [52].

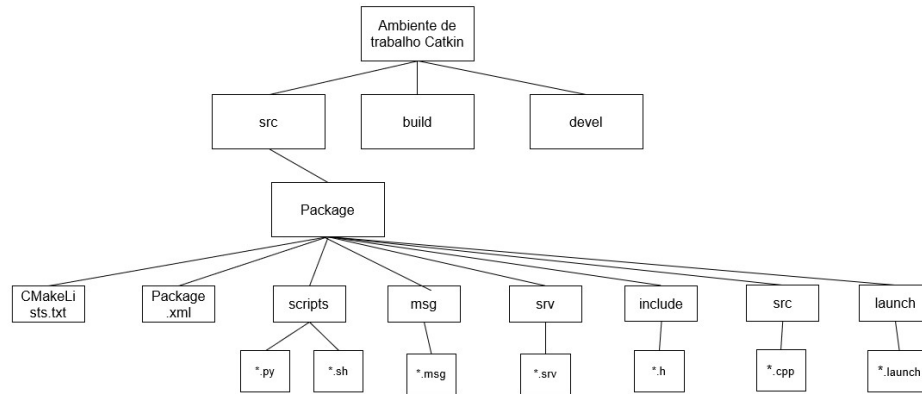


Figura 36 – Sistema de arquivos do ROS. (Baseado em [54])

Como já foi referido, um nó pode realizar uma publicação ou infinitas subscrições, no entanto antes de inicializar um *nó*, existe um pré-requisito no ROS que é o funcionamento de um *nó* mestre denominado por ROScore. O ROScore é responsável por iniciar a conexão de mensagens entre nós [53].

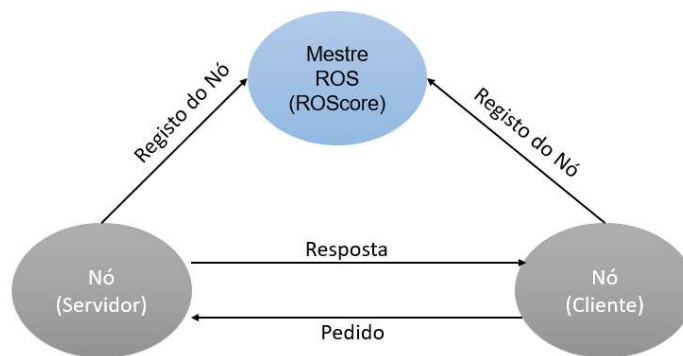


Figura 37 – Funcionamento ROScore. (Baseada em [55])

Os tópicos consistem na forma básica de comunicação entre nós, onde através de scripts/programas podem enviar ou receber mensagens, como se pode observar na figura 38. Este método de comunicação é baseado num mecanismo de *publisher/subscriber*, onde define um programa que publica mensagens para um tópico e outros programas que realizam a subscrição desse tópico. Para tal tanto o *publisher* como o *subscriber* devem de ter na sua constituição o mesmo tipo de mensagem para que seja possível realizar a comunicação entre eles [52].

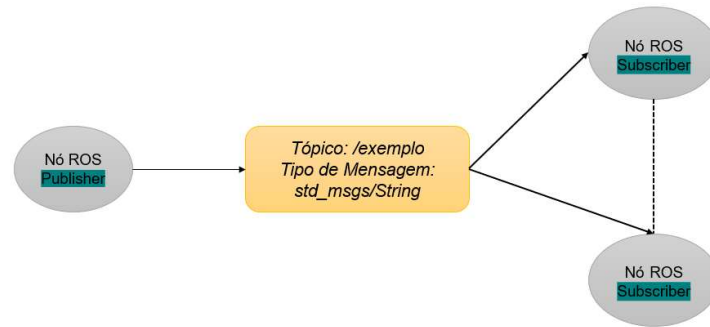


Figura 38 – Exemplo da comunicação de um tópico [56]

CAPÍTULO 3: Metodologia Proposta

Neste capítulo descreve-se a modelação do sistema sensorial Sentry. Será discutido todo o processo que envolveu a implementação deste sistema, desde os métodos e os respetivos conceitos teóricos, seleção dos componentes usados, e a aplicação de filtros e algoritmos.

3.1 Estrutura Mecânica

O sistema chamado Sentry, desenvolvido neste projeto, foi concebido como uma solução "*plug and play*", adaptada às especificações de cada máquina florestal. Ele incorpora todos os sensores necessários para permitir diferentes níveis de autonomia na máquina. Para integrar o sistema, basta conectar uma ficha de 64 pinos que é comum em tratores da MDB. A estrutura mecânica do Sentry é composta por uma caixa de alumínio projetada para conter todos os componentes necessários para o seu funcionamento adequado. A Figura 39 ilustra as partes constituintes que compõem o Sentry.

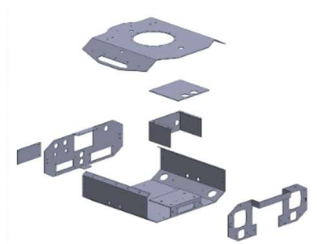


Figura 39 – Partes constituintes do Sentry

A Figura 40 exibe a incorporação de todos os componentes do sistema Sentry.

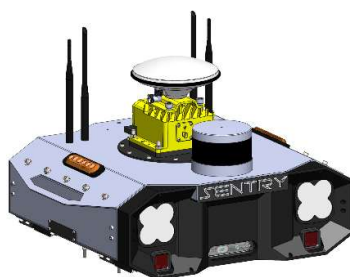


Figura 40 – Final design do Sentry

O Sentry foi instalado na parte traseira da máquina, usando para o efeito uma estrutura metálica que foi montada com um suporte elevatório composto por duas placas de contraplacado marítimo e varões roscados. Essa configuração eleva o Sentry, permitindo que as câmaras tenham um amplo campo de visão, *Field of View* (FOV) durante todas as operações de trabalho. A Figura 41 (a), apresenta o campo de visão FOV abrangido pelas câmaras, enquanto que a Figura 41 (b), mostra a disposição do Sentry após a sua montagem no MDB LV600.



Figura 41 – Integração Mecânica do Sentry na LV 600 PRO: (a) Verificação dos Campos de Visão das Câmaras; (b) Validação da Estrutura Mecânica na LV 600 PRO

3.2 Arquitetura do Hardware

O Sentry foi concebido para realizar uma variedade de tarefas, que podem ser divididas em duas categorias principais: tarefas fundamentais e tarefas funcionais. As tarefas fundamentais envolvem a compreensão do comportamento da máquina durante a sua operação, como verificação da posição, inclinação e identificação de objetos. As tarefas funcionais dizem respeito às ações que o Sentry pode executar, como mover as lagartas, ativar o triturador e controlar a elevação ou descida da ferramenta, entre outras. Para executar essas tarefas, o Sentry é equipado com sistemas de aquisição e processamento de dados.

Esta subsecção está subdividida em duas partes. Na primeira parte, é descrito o processo de interpretação dos sinais provenientes da ficha de 64 pinos que se encontra no trator. Na segunda parte, é apresentado o *hardware* que será integrado no Sentry, incluindo as câmaras, o sistema de controlo remoto, a fonte de alimentação, o microcontrolador utilizado e o sistema de locomoção.

3.2.1 Compreensão dos sinais da máquina

O primeiro passo passou por interpretar os sinais que são enviados para a ficha de 64 pinos. A posição de cada pino da ficha pode ser conhecida através do esquema fornecido pelo fabricante. Para a realização deste passo, foram feitas medições da tensão em cada pino da ficha, com o motor da máquina desligado e posteriormente foram enviados sinais através do controlo remoto, por sinais rádio, registando-se o valor de tensão recebido. Desta forma, foi possível compreender a funcionalidade de cada pino da ficha com o intuito de efetuar a escolha do *hardware*.

A Figura 42 ilustra o processo de operação da máquina quando é utilizado o controlo remoto. Por meio de um dispositivo rádio de controlo remoto, enviam-se sinais de radiofrequência para um módulo instalado na máquina, o qual converte esses sinais em comandos de movimentos específicos da máquina florestal.



Figura 42 – Método de operação da máquina utilizando o controlo remoto

3.2.2 Escolha do Hardware

Com base nas medições realizadas na ficha de 64 pinos, foi possível determinar o *hardware* necessário para a construção do sistema sensorial do Sentry. A arquitetura de *hardware* do Sentry foi projetada seguindo uma abordagem de um sistema distribuído, como ilustrado no diagrama de blocos apresentado na Figura 43. Neste projeto, o computador desempenha o papel central na arquitetura, funcionando como o cérebro do sistema. Os próximos tópicos apresentam todos os equipamentos inseridos no Sentry, e realizam uma análise comparativa entre os equipamentos.

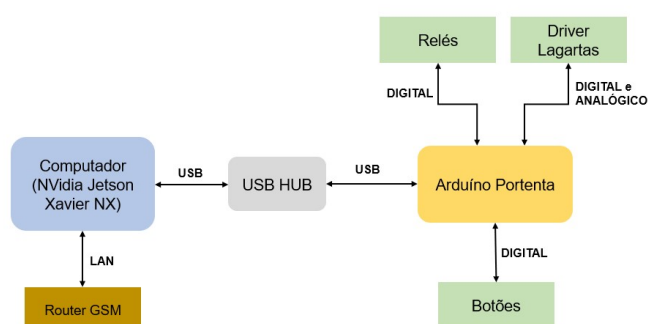


Figura 43 – Arquitetura de Hardware do Sentry

1) Computador

O sistema Sentry é equipado com um computador responsável por gerenciar todos os processos, desempenhando o papel de cérebro do sistema. A seleção desse computador envolveu um processo criterioso no qual foram comparados dois computadores, uma vez que era necessário encontrar um dispositivo com elevada capacidade de processamento, baixo consumo de energia e dimensões compactas. Devido a essas restrições de tamanho, a escolha recaiu sobre dois minicomputadores: a NVIDIA Jetson Xavier Nx e o Intel NUC 1185G7 (conforme mostrado na Figura 44). A Tabela que se encontra no Anexo A apresenta uma comparação entre esses dois equipamentos resultante da informação disponível em [57,58].

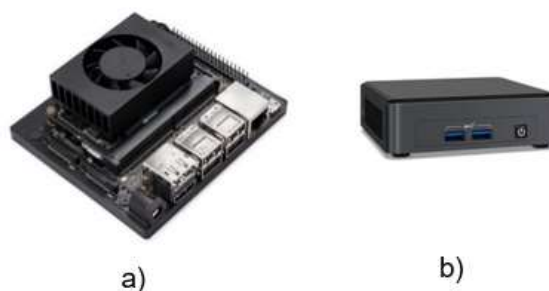


Figura 44 – Minicomputadores escolhidos para o Sentry. a): Nvidia Jetson Xavier Nx; b): Intel NUC 1185G7

Após uma análise comparativa com base na Tabela do Anexo A, verifica-se que a NVidia Jetson Xavier Nx se destaca como a escolha mais adequada para desempenhar o papel de cérebro no Sentry. A NVidia Jetson possui características notáveis que permitem a execução eficiente de algoritmos, incluindo redes neurais, em simultâneo com o processamento de dados provenientes de sensores de alta resolução. Além disso, este minicomputador possui um baixo consumo de energia e dimensões compactas. Embora a NVidia Jetson Xavier Nx seja ligeiramente mais cara do que o Intel NUC 1185G7, ela revela-se a solução ideal para sistemas que requerem o uso de Inteligência Artificial, bem como para aplicações de robótica e sistemas com sensores de alta resolução [57].

A NVidia Jetson Xavier Nx vai servir de interface global para todos os dispositivos sensoriais tais como as câmaras, o sistema GPS, o IMU, o LIDAR e também com o microcontrolador que vai ser responsável por controlar as ações da máquina. A NVidia Jetson Xavier Nx está localizada dentro do Sentry.

2) Router 3G / 4G

O Sentry é projetado para viabilizar a comunicação remota entre o sistema e um usuário por meio de uma aplicação web. Para garantir que o estado da máquina seja monitorado durante sua operação, é essencial estabelecer uma conexão de rede móvel, permitindo que o computador permaneça sempre conectado à Internet para a troca de informações. Portanto, é necessário equipar o Sentry com um router móvel adequado.

A escolha deste *router* deve considerar o suporte à tecnologia 4G LTE, velocidades rápidas de *upload* e *download*, recursos de segurança robustos e dimensões compactas. Após uma pesquisa de mercado, foram identificados dois possíveis candidatos: o RUT360 da Teltonika e um modelo da D-Link (Figura 45). A Tabela presente no Anexo B apresenta uma comparação detalhada entre estes dois routers.



Figura 45 – Routers 3G/4G escolhidos para o Sentry: (a) RUT360; (b) D-link DWR-921

Ao analisar as especificações listadas na Tabela do Anexo B, é notório que o router RUT360 oferece um conjunto mais sólido de recursos em comparação com o DWR-921. Esses recursos incluem uma velocidade de *download* superior, dimensões compactas, baixo peso e eficiência energética. Com base nesses critérios, a escolha recaiu no router RUT360. Este router é projetado especificamente para atender às necessidades de aplicações industriais e comerciais, destacando-se pela sua confiabilidade e capacidade de se conectar a diversas redes. O RUT360 é altamente versátil, oferecendo compatibilidade com várias tecnologias de rede, como redes móveis 4G LTE, redes Wi-Fi sem fio e conexões Ethernet cabladas. Além disso, ele é capaz de suportar velocidades de transferência de dados elevadas, tanto para *download* quanto para *upload*, proporcionando acesso à Internet de alta velocidade [59]. O RUT360 foi instalado/montado dentro do Sentry.

3) Sistema de controlo da máquina florestal

O Sentry incorpora um sistema de controlo centralizado responsável por ativar as funcionalidades do trator conforme necessário. Nesse contexto, o Sentry possui um controlador centralizado principal e um controlador centralizado de assistência, sendo este a Nvidia Jetson Xavier Nx e um microcontrolador Arduino Portenta, respetivamente. A escolha do Arduino Portenta foi motivada pela sua alimentação via conexão USB, simplificando a rede de conexões, pelo seu tamanho compacto e pela compatibilidade das suas bibliotecas com o ROS, o que torna a comunicação com o controlador central principal muito eficaz.

O Arduino Portenta, atuando como o controlador centralizado de assistência, colaborará com o sistema principal na execução de determinados comandos para controlar os movimentos do trator. Além do sistema de assistência, diversos outros componentes eletrónicos estão presentes, incluindo relés de *Low Level Trigger* (LLT) e o *driver* (DRI0041 da DF-Robot), que assumirão a responsabilidade pelos movimentos da máquina florestal. A utilização destes componentes deriva das limitações de corrente e tensão no sistema de controlo de assistência, que não são compatíveis com as entradas do trator. Este sistema de controlo encontra-se instalado dentro do Sentry.

4) Sistema RTK

O sistema Sentry está equipado com um sistema *Real-Time Kinematic* (RTK) que serve para melhorar o posicionamento do sistema GNSS. Como referido na secção 2.4 o sinal GNSS no percurso entre os satélites e o recetor é afetado pela atmosfera, por obstáculos no cenário podendo os sinais sofrer desvanecimento multipercurso causando erro na localização que, em média, podem variar entre os 2 a 4 metros [60].

Em ambientes florestais, a resolução mencionada anteriormente não atende aos requisitos necessários. Para aprimorar a precisão do sistema de navegação, recorre-se ao sistema RTK, que é um método de correção de posição projetado para sistemas de navegação por satélite GNSS. Um sistema RTK compreende duas unidades principais: uma estação base e uma unidade instalada no robô. A estação base permanece estacionária, enquanto a unidade no robô é montada na plataforma robótica. A estação base transmite informações de posição em tempo real para o robô, permitindo assim a correção da sua localização, aumentando dessa forma a precisão dos valores obtidos [60]. No trabalho de *Ryan Moeller et al.* em [60], foi realizada uma comparação entre quatro sistemas RTK: Piksi Multi Evaluation Kit, Trimble R8s, Spectra Precision SP80 e Geomax Zenith35 Pro, onde o *Piksi Multi Evaluation Kit* para além de ser o mais económico, apresenta um melhor desempenho na correção da posição GPS, desta forma optou-se por este sistema RTK. Neste caso a unidade do robô vai estar implementada dentro do Sentry e a unidade da estação base num local onde seja possível abranger a área de trabalho do robô.

5) Sistema de alimentação

O sistema de alimentação do Sentry é composto por uma bateria de LIFEP04 com 12,8 V tensão e capacidade de 48 Ah. Além dessa bateria, o sistema também integra dois conversores: um deles disponibiliza uma saída de 12 V para alimentar todos os componentes sensoriais e computacionais, enquanto o segundo fornece 12 V específicos para alimentar os controladores das lagartas. Este sistema encontra-se situado dentro do Sentry.

6) Sistema de botões

O sistema Sentry está equipado com três botões, cada um com funções distintas:

- Botão de emergência, tipo cogumelo, que, quando pressionado, interrompe a alimentação dos controladores das lagartas e envia informações ao controlador de assistência. Isso permite que o controlador de assistência saiba o estado do sistema e desligue a ignição da máquina, se necessário;
- Botão de inicialização de todos os sistemas. Este botão possui dois estados: um estado que liga a parte sensorial e o *hardware* e outro estado que desliga completamente o Sentry;

- Botão de ignição do trator. Este botão possui apenas um estado, pois seu propósito é enviar um impulso para iniciar a ignição da máquina.

Esses botões estão localizados na parte traseira do Sentry, conforme ilustrado na Figura 48.

7) Sistema de fusíveis

O sistema Sentry incorpora um mecanismo de segurança contra curtos-circuitos, implementado por meio do uso de fusíveis. Desta forma, foram dimensionadas as correntes máximas suportadas pelos sensores e *hardware* e estipularam-se valores de correntes para cada fusível. Os fusíveis estão localizados numa caixa de fácil acesso, localizada na parte traseira do Sentry, de modo a simplificar a sua substituição quando necessário, como se pode observar na Figura 48.

3.3 Arquitetura Sensorial

O Sentry está equipado com uma ampla variedade de sensores, com o objetivo de alcançar redundância durante sua operação e, ao mesmo tempo, assegurar uma maior eficiência do sistema. Dentro do Sentry, são integrados três módulos de sensores distintos:

- Módulo GPS+IMU+MAG: Este módulo permite a obtenção da posição e orientação do robô, combinando informações de GPS, com uma unidade inercial (IMU) e um magnetómetro;
- Módulo de Sensor Laser 3D: Este módulo possibilita a varredura da área de operação do trator, fornecendo informações detalhadas sobre o ambiente onde o Sentry está a operar;
- Módulo de Sistema de Visão: Este módulo é composto por duas câmaras RGBD e duas câmaras térmicas, permitindo a obtenção de informações detalhadas do ambiente de operação.

A Figura 46 ilustra as interfaces utilizadas e os fluxos de sinal na arquitetura atual dos sensores adotados para o Sentry.

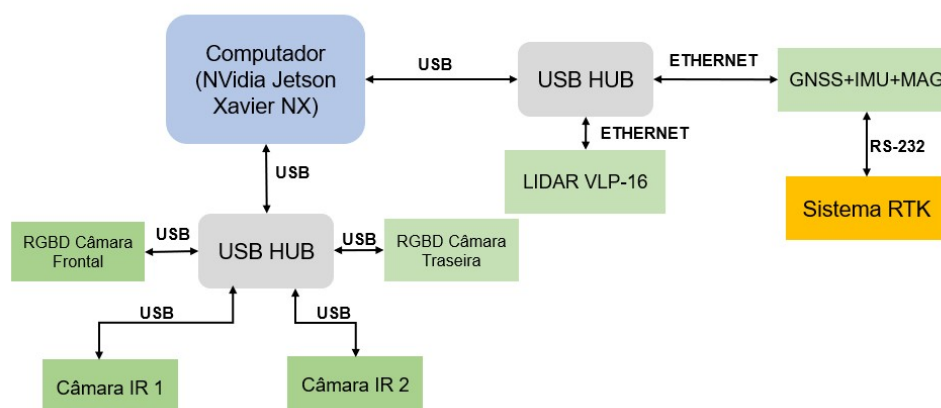


Figura 46 – Arquitetura de sensores do Sentry

1) GPS+IMU+MAG

Para determinar a posição e a orientação da MDB LV600 no Sentry, é necessário um sistema com um dispositivo capaz de adquirir dados de GPS, IMU e magnetômetro. Dado que o Sentry segue uma abordagem "*plug and play*", está fora de opção a instalação de *encoders* nas lagartas do trator, logo é necessário desenvolver um sistema que disponibilize informações de orientação e posição com elevada precisão. Em resposta a essas exigências, foi selecionado o sistema *Duro Inertial*, representado na Figura 47. Este é um receptor GNSS de dupla frequência, cujo custo é elevado, mas que atende aos requisitos de integração deste componente no Sentry.



Figura 47 – Duro Inertial

O sistema *Duro Inertial* da *Swift Navigation* combina tecnologia GNSS com RTK para aprimorar a precisão do posicionamento. Ele usa o GNSS padrão, incluindo GPS e GLONASS, e recebe correções em tempo real para alta precisão. Além disso, possui um IMU Bosch BMI160 com 6 DoF para medição de aceleração e taxa de rotação, tornando o posicionamento mais preciso em ambientes complexos. Também inclui um magnetômetro Bosch BMM150 de 3 eixos com alta precisão na detecção de campos magnéticos fracos e fortes. Essas características contribuem para estimativas precisas do estado do robô [61]. Com a integração do sistema *Duro Inertial* é possível localizar a posição e a orientação do trator durante a sua operação.

2) Sistema de visão

Para a navegação autónoma segura de máquinas florestais, como a MDB LV600 PRO, é imperativo considerarem-se questões de segurança, incluindo a deteção de pessoas e outros obstáculos. Para atender a essa necessidade, o Sentry foi equipado com um sistema de visão composto por quatro câmaras, sendo duas delas câmaras RGBD e as outras duas câmaras térmicas. No que diz respeito às câmaras RGBD, duas opções foram consideradas: a Intel Realsense D435i e a ZED 2i. Essa seleção foi baseada num estudo comparativo realizado por Vladimir Tadic et al., conforme documentado em [62], e descrito na Tabela do Anexo C que apresenta uma análise detalhada dessas câmaras.

Com base nas informações fornecidas na Tabela do Anexo C, a câmara ZED 2i apresenta um desempenho superior em termos de recursos e funcionalidades. No entanto, devido à fase de desenvolvimento em que o sistema Sentry se encontra, foi escolhida a câmara Intel Realsense D435i, que oferece um custo mais acessível em comparação com a ZED 2i.

A Intel Realsense D435i é uma câmara de deteção de profundidade projetada para diversas aplicações, incluindo visão computacional. Destaca-se pelo seu amplo campo de visão de $91^\circ \times 65^\circ \times 100^\circ$ para a câmara RGB, o que minimiza pontos cegos no mapa de profundidade. O FOV do sensor de profundidade correspondente é de $85^\circ \times 58^\circ \times 90^\circ$. A notável precisão na medição de profundidade da D435i torna-a adequada para uma ampla diversidade de aplicações. Com um alcance de deteção de profundidade de até 10 metros, torna essa câmara a solução apropriada para uma ampla gama de aplicações [62].

Em relação às câmaras térmicas foi escolhida a FLIR ADK, desenvolvida pela FLIR System. Este sistema destaca-se por fornecer informações mesmo em condições de visibilidade reduzida, tais como: escuridão, neblina, nuvens de fumo ou condições climáticas adversas comuns [63].

Além disso, a FLIR ADK pode ser integrada com outros sensores, como câmaras RGB e sensores LIDAR, para criar um sistema de visão abrangente do ambiente ao redor da máquina. Essa combinação de sensores aprimora a percepção do cenário, permitindo a identificação de objetos e melhorando a segurança durante a navegação do veículo [63]. A necessidade de integrar câmaras FLIR ADK deve-se a situações adversas, como poeira, que podem ocorrer durante a limpeza em áreas florestais. Em ambientes florestais, a introdução de poeira é comum e pode afetar o desempenho de sensores das câmaras RGBD, tornando as câmaras infravermelhas uma adição valiosa para essas situações [64].

No Sentry, as câmaras RGBD serão posicionadas na parte frontal e traseira da caixa, enquanto as câmaras térmicas estarão localizadas na parte frontal.

3) Sensor Laser 3D

De forma a perceber o ambiente no qual a plataforma robótica opera, o Sentry incorpora também com o sensor 3D LiDAR Velodyne Puck VLP-16.

O sensor Laser Velodyne VLP-16 é uma escolha ideal para aplicações de robótica, graças ao seu design compacto, confiabilidade e excelente desempenho. Com aproximadamente 16 canais de laser, ele permite uma visão de 360 graus, proporcionando uma compreensão abrangente do ambiente, alta resolução espacial para detetar objetos com grande detalhe e a capacidade de segmentar e separar imagens [65].

No Sentry, o sensor 3D LiDAR foi posicionado na parte superior da caixa, conforme representado na Figura 48.

3.4 Montagem do Sentry

Após a escolha dos diversos componentes que fazem parte do sistema Sentry, procedeu-se à conexão dos mesmos. Na Figura 48 são perceptíveis a disposição e a montagem de todos os elementos que formam o sistema Sentry.



Figura 48 – Demonstração de todos os sensores montados no Sentry

3.5 Montagem do Sistema RTK

Como referido na secção 3.2 o sistema RTK necessita de ter uma unidade na estação base e outra no Sentry. Observando a Figura 49, é descrito o processo de funcionamento do sistema RTK. Neste caso a estação base encontra-se em cima de um suporte não amovível, com o objetivo de a sua posição permanecer inalterada ao longo do tempo, com uma antena direcional de 7.5 dBi.

Do lado do Sentry uma antena omnidirecional vai captar os sinais rádio frequência transmitido da estação base.

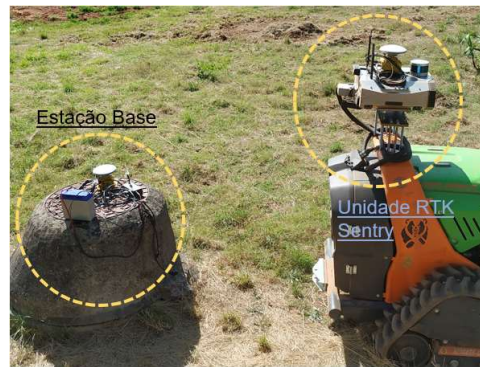


Figura 49 – Princípio funcionamento do Sistema RTK com o Sentry

3.6 Integração com o ROS

Para a integração do ROS no Sentry foi instalado esta *framework* no computador principal, a NVidia Jetson Xavier Nx. De seguida instalaram-se todas as *packages* associadas aos sensores existentes no Sentry, assim como a comunicação com o controlador Arduino Portenta. Na Tabela 3 estão apresentadas todas as *packages* associadas aos sensores e a referência à origem do código fonte utilizado.

Tabela 3 – Package ROS para comunicar com os sensores/componentes do Sentry

Sensor / componente	Nome da Package	Código Fonte
Duro Inertial	duro_gps_driver	Código fonte disponível em [66]
Velodyne VLP-16	velodyne	Código fonte disponível em [67]
Arduino Portenta	Rosserial	Código fonte disponível em [68]

Com base na Tabela 3, a Package do *Duro Inertial* fornecerá tópicos relacionados aos dados do IMU, GPS e Magnetómetro. A Package do Velodyne VLP-16 permitirá a aquisição de dados da nuvem de pontos do LIDAR. A Package do Arduino desempenhará um papel fundamental na criação de comunicação por meio de tópicos ROS entre o computador principal e o controlo de assistência. Dessa forma, o Arduino estará capacitado para interpretar mensagens do ROS a fim de controlar o sistema de locomoção da máquina.

A partir daqui é possível manipular os valores para que estes vão de encontro ao objetivo deste projeto. Os *scripts* realizados para colaborar com a *framework* ROS utilizaram a linguagem de programação Python3.

3.7 Aquisição de Orientação

A aquisição da orientação do robô em relação a um referencial fixo tem por finalidade calcular o *heading*, que está relacionado à rotação do corpo ao redor do eixo vertical (eixo Z), na forma de um vetor que representa a direção do robô. Esse parâmetro é essencial para permitir que o robô se mova ao longo de uma trajetória específica. Neste caso, a obtenção do *heading* é baseada nos dados do

magnetômetro provenientes da *package* do ROS. Essa abordagem tem sido amplamente utilizada e, quando combinada com giroscópios e acelerômetros, proporciona dados mais confiáveis [69]. Para obter uma estimativa precisa do *heading*, é necessário seguir um conjunto de procedimentos, que podem ser observados no diagrama de blocos apresentado na Figura 50.

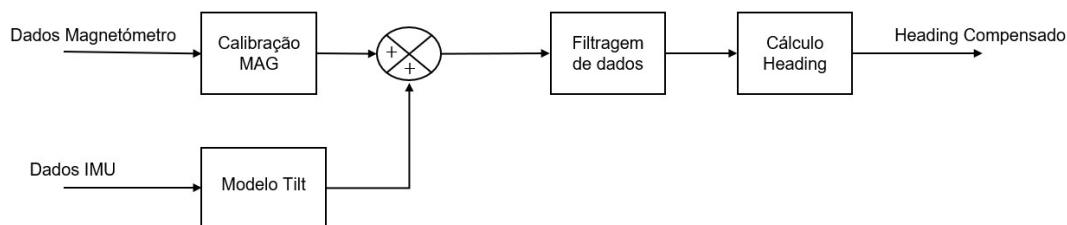


Figura 50 – Diagrama de blocos para obtenção do Heading

Seguindo o diagrama de blocos da Figura 50, para obtenção do *heading* é necessário:

- a) Calibrar o magnetômetro;
- b) Filtrar os dados;
- c) Compensar o *tilt*;
- d) Calcular o *heading*.

As próximas subsecções vão analisar os diferentes blocos do diagrama da Figura 50, que representam todos os passos idealizados para a obtenção do *heading*.

3.7.1 Calibração do Magnetômetro

Como já referido o princípio do magnetômetro consiste em medir o campo magnético numa dada região, que por sua vez constitui uma componente paralela à superfície da Terra que aponta sempre para o Norte magnético. Devido ao facto de os magnetômetros utilizados em veículos em movimento geralmente não estarem fixos à superfície da Terra, é de extrema importância empregar magnetômetros de três eixos montados de maneira ortogonal. Isto permite a realização de uma rotação completa do campo magnético terrestre, ajustando-o de volta a uma orientação horizontal, conforme ilustrado na Figura 51. Neste contexto, o magnetômetro é alinhado com o robô (no caso específico, alinhado com o Sentry, que por sua vez será alinhado com a MDB LV600). O eixo X coincide com a direção de movimento frontal do veículo, o eixo Z aponta na vertical, e o eixo Y estende-se na direção do movimento transversal do veículo, perpendicular ao plano formado pelos eixos X e Z. Os 3 ângulos de Euler: *roll* (γ), *pitch* (β) e *yaw/heading* (α), correspondem aos ângulos de rotação dos eixos X, Y e Z, respetivamente. Visto que é possível conhecer os ângulos *pitch* e *roll*, consegue-se projetar num plano horizontal o vetor do campo magnético que fique paralelo à superfície da terra conforme a seguinte equação matricial [69]:

$$\begin{bmatrix} M_{xH} \\ M_{yH} \end{bmatrix} = \begin{bmatrix} \cos(\beta) & \sin(\beta)\sin(\gamma) & -\cos(\gamma)\sin(\beta) \\ 0 & \cos(\gamma) & \sin(\gamma) \end{bmatrix} \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} \quad (23)$$

Os parâmetros M_x , M_y e M_z são os dados provenientes do magnetómetro na *frame* do robô e M_{xH} e M_{yH} são os dados resultantes do magnetómetro na *frame* do robô projetada no plano horizontal (X_h e Y_h conforme mostra a Figura 51 [69]).

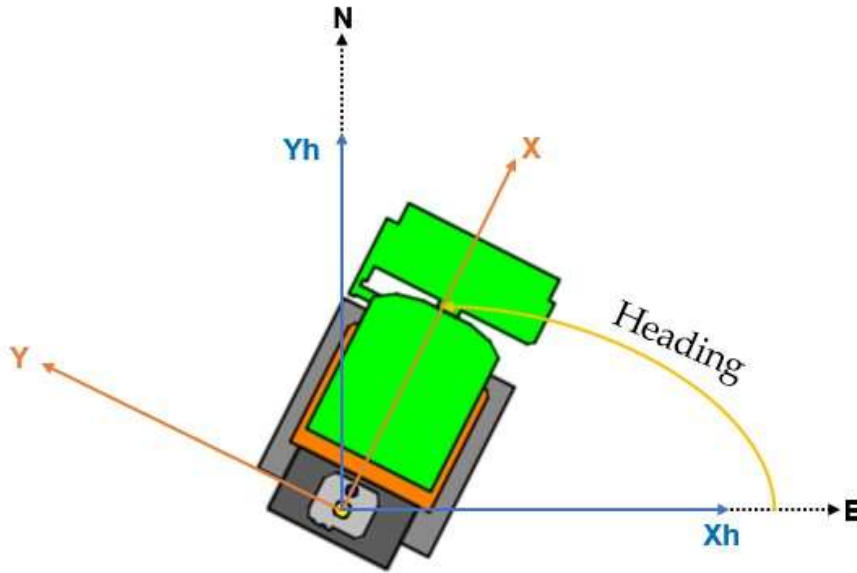


Figura 51 – Determinação do Heading com um magnetómetro de 3 eixos

No entanto dados provenientes de um magnetómetro podem ser influenciados por interferências que estão divididas em dois grupos:

- Hard iron: esta categoria engloba campos magnéticos que são estáticos ou apenas levemente variáveis ao longo do tempo e onde a sua origem é proveniente de materiais ferromagnéticos [70].
- Soft iron: refere-se ao campo magnético gerado internamente pelo próprio dispositivo [70].

Portanto, a calibração assume um papel crítico, e é essencial desenvolver um modelo para a calibração do magnetómetro, no qual são considerados parâmetros como o *Bias* e o Fator de Escala, que podem ser representados da seguinte maneira [70]:

$$\hat{M}_{xH} = SF_{xH} \times M_{xH} + B_{xH} \quad (24)$$

$$\hat{M}_{yH} = SF_{yH} \times M_{yH} + B_{yH} \quad (25)$$

$$\hat{M}_{zH} = SF_{zH} \times M_{zH} + B_{zH} \quad (26)$$

onde $\hat{M}_{xH}$, $\hat{M}_{yH}$ e $\hat{M}_{zH}$ correspondem aos valores reais da projeção do campo magnético num plano horizontal. SF_{xH} , SF_{yH} e SF_{zH} são os fatores de escala e erros de medição do magnetómetro e B_{xH} , B_{yH} e B_{zH} corresponde à aplicação do *Bias* nas medições do magnetómetro.

Finalmente para obter o *heading* é necessário realizar o arco tangente 2 com base nos valores obtidos:

$$Heading = atan2(\hat{M}_{yH}, \hat{M}_{xH}) \text{ radians} \quad (27)$$

ou em alternativa:

$$Heading = atan2(\hat{M}_{yH}, \hat{M}_{xH}) \times \frac{180}{\pi} \text{ degrees} \quad (28)$$

Portanto, é essencial desenvolver um algoritmo que seja capaz de adquirir dados provenientes do ROS e realizar os cálculos necessários para obter o *heading*. A Figura 52 apresenta um fluxograma que abrange todo o processo, desde a aquisição de dados até a obtenção do *heading*. O processo de aquisição envolve a rotação do magnetómetro em torno de todos os eixos, resultando na recolha de aproximadamente 1000 pontos em cada eixo. Após essa etapa, os parâmetros de fator de escala e Bias são calculados, aplicando as equações 24 a 26.

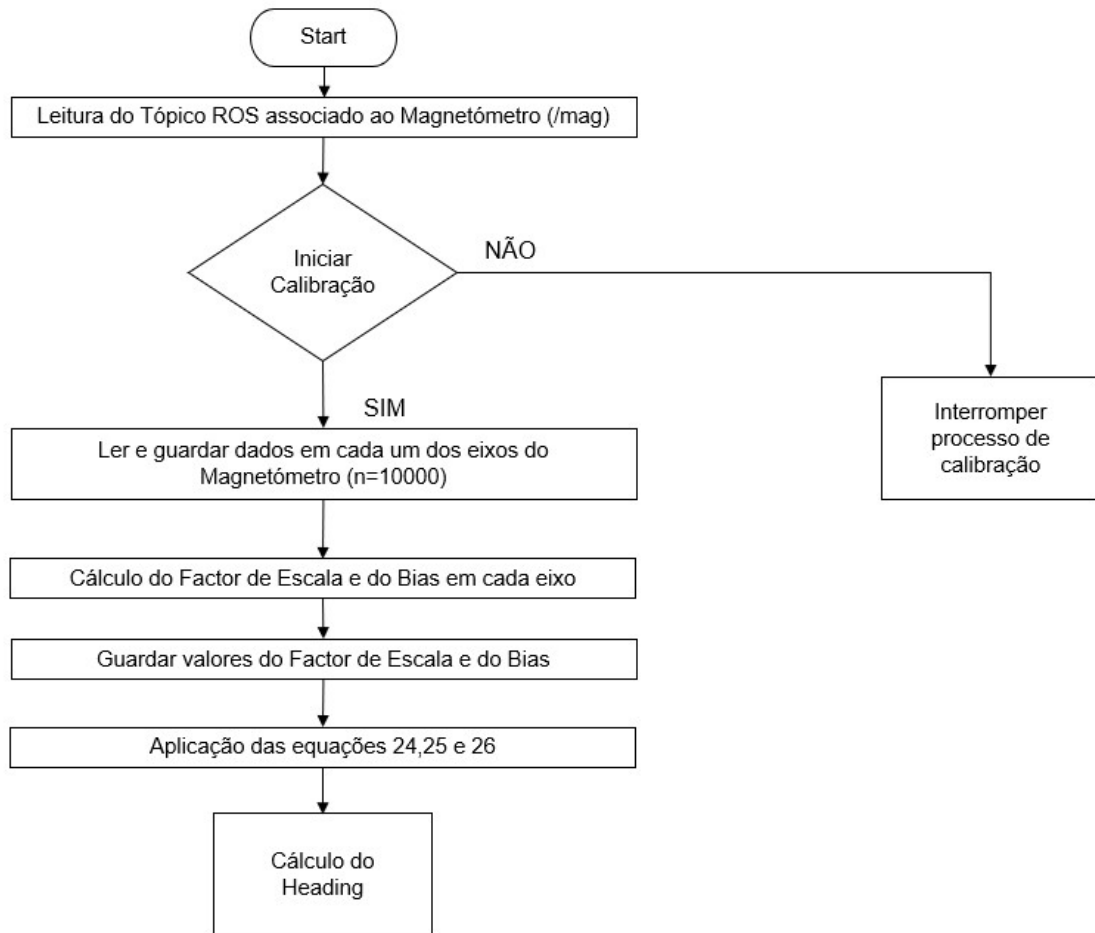


Figura 52 – Processo para a calibração do magnetómetro

3.7.2 Filtragem de dados

Os dados provenientes do magnetómetro apresentam ruído, sendo este designado por *pink noise* ou ruído $1/f$ às baixas frequências e *white noise* às altas frequências [71]. Por esta razão, optou-se por adotar duas estratégias de filtragem. Isso inclui a implementação de um filtro passa-baixo, *Low-Pass Filter* (LPF) e um filtro de média exponencial móvel *Exponential Moving Average Filter* (EMAF) com o propósito de minimizar o ruído.

No que concerne ao filtro passa-baixo este consiste num filtro do tipo IIR e antes de proceder à implementação deste é necessário entender o processo de amostragem onde se converte um sinal analógico contínuo para um sinal digital discreto [40]. A taxa de amostragem é denominada por F_s e segundo o teorema de Nyquist a taxa de amostragem deve obedecer o seguinte critério [40]:

$$F_s \geq 2 \times F_c \quad (29)$$

Onde, F_s representa a taxa de amostragem (em Hz) e F_c é a frequência máxima (em Hertz). Assim garante-se que não há perdas de informações durante o processo de amostragem evitando *aliasing* e permitindo a aplicação adequada do filtro [40].

Desta forma foi desenvolvido um algoritmo que recolheu os dados previamente calibrados do magnetómetro e aplicou-se um filtro passa-baixo como descrito no fluxograma da Figura 53, que descreve o algoritmo do procedimento representado. Primeiramente, inicia-se o processo verificando a calibração do magnetómetro. Se o magnetómetro estiver calibrado, isso significa que os fatores de escala e o *Bias* já foram previamente calculados, permitindo avançar para a escolha da ordem do filtro, a qual influencia a resposta em frequência do filtro [40]. Após estes procedimentos, é executada a filtragem dos dados utilizando a equação 16, considerando valores de frequência de corte inferiores à frequência de Nyquist. Finalmente, conclui-se esse processo com o cálculo do *heading*.

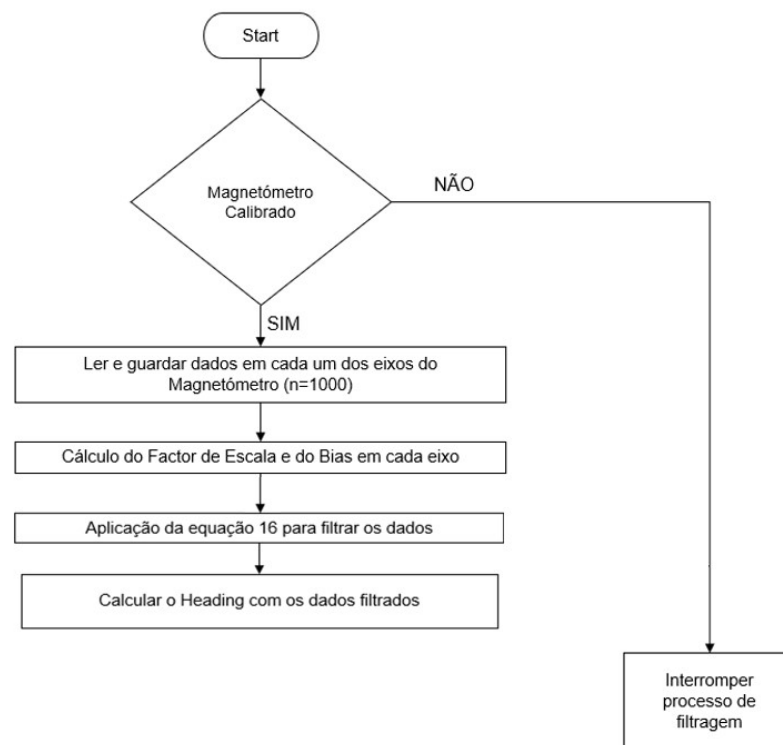


Figura 53 – Processo para aplicação de um LPF aos dados calibrados do magnetómetro

A segunda abordagem consistiu em usar um filtro do tipo média móvel. Conforme descrito na secção 2.7 existem 4 tipos de *Moving Average Filters* sendo que foi escolhido para este trabalho o EMAF, devido ao facto de dar mais peso aos valores mais recentes [44], sendo estes armazenados numa *double-ended queue (deque)*, que consiste numa lista/array que realiza a inserção e a remoção de dados de forma mais rápida. A equação que representa o EMAF é definida da seguinte forma:

$$EMA = \frac{\sum_{i=0}^{n-1} (i+1)^2 \times x[i]}{\sum_{i=0}^{n-1} (i+1)^2} \quad (30)$$

onde:

- $x[i]$ representa o valor do elemento no índice i numa lista;
- $(i + 1)^2$ é o peso dado a cada elemento, que diminui exponencialmente com o aumento do índice i .

Na equação 30, o numerador é a soma ponderada dos elementos da lista, e o denominador é a soma dos pesos, calculados com base no índice de cada elemento. Os pesos diminuem exponencialmente à medida que o índice aumenta, e desta forma a média móvel exponencial responde enfatizando os valores das amostras mais recentes. Para implementar a EMAF na filtragem dos dados do magnetómetro, é usada uma *deque* para colocar os dados por ordem cronológica, sendo a dimensão dessa *deque* fixa, onde posteriormente um peso exponencial é associado aos dados recolhidos conforme se mostra na Figura 54.

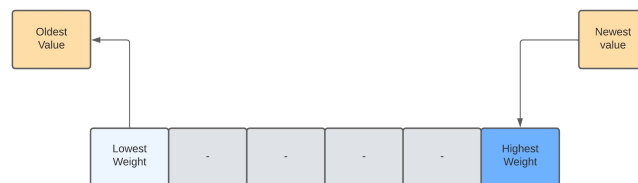


Figura 54 – Exemplo funcionamento *deque* para o EMAF

O diagrama de blocos representado na Figura 54 ilustra o funcionamento básico do EMAF, aplicado aos dados do magnetómetro:

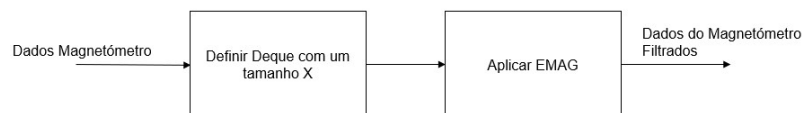


Figura 55 – Diagrama de blocos para aplicação do EMAF

O fluxograma que descreve a operação deste filtro está presente na Figura 56. Inicialmente, são inseridos valores numa *deque* com um tamanho específico. Quando a fila está cheia, o filtro de média móvel exponencial - EMA é aplicado, resultando nos dados do magnetómetro filtrados.

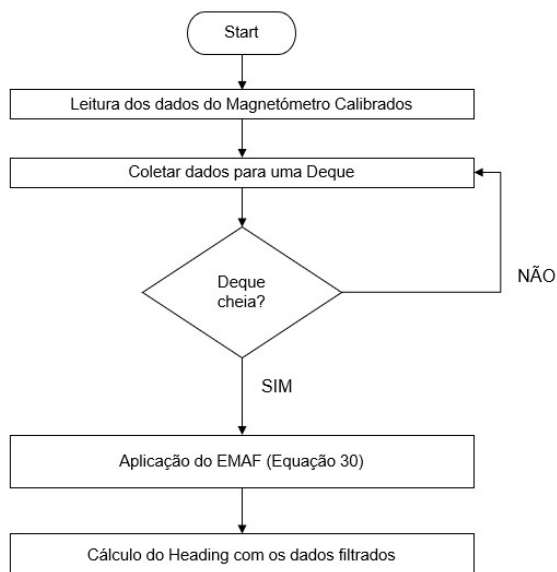


Figura 56 – Aplicação de um EMAF aos dados calibrados do magnetômetro

3.7.3 Compensação do *tilt*

A compensação do *tilt* (inclinação) do magnetômetro é um processo importante para garantir leituras precisas das posições angulares em relação ao campo magnético da Terra. O *tilt* refere-se ao ângulo entre o plano de referência do magnetômetro e a direção vertical da Terra. Ao utilizar as leituras de um acelerômetro e/ou giroscópio, é possível determinar os ângulos de inclinação (*roll* e *pitch*), os quais podem ser utilizados para corrigir as leituras do magnetômetro [70].

Na Figura 57 está representada a relação entre o sistema de coordenadas do corpo do *Duro Inertial* com os parâmetros X_b , Y_b e Z_b e os eixos X , Y e Z correspondentes à *frame* do acelerômetro, do giroscópio e do magnetômetro.

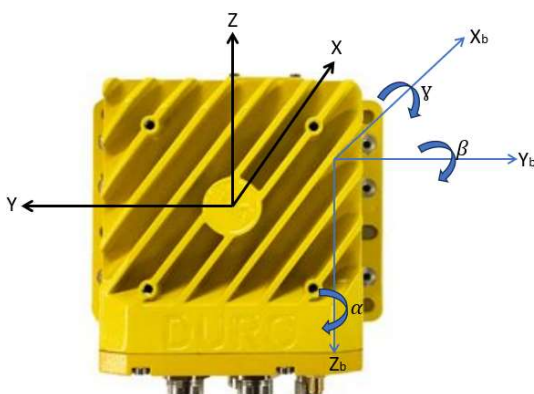


Figura 57 – Frames Duro em relação ao Acelerômetro, Giroscópio e Magnetômetro

As rotações ilustradas na Figura 57 correspondem aos ângulos de Euler *roll* (γ) - *pitch* (β) - *heading* (α). Esses ângulos podem ser definidos utilizando matrizes de rotação, como descrito nas Equações 1, 2 e 3. Consequentemente, as fórmulas para compensar a inclinação nos eixos X e Y são apresentadas nas Equações 31 e 32 [70]:

$$X_H = M_x \times \cos(\beta) + M_z \sin(\beta) \quad (31)$$

$$Y_H = M_x \times \sin(\gamma) \times \cos(\beta) + M_y \times \cos(\gamma) - M_z \sin(\gamma) \times \cos(\beta) \quad (32)$$

onde:

- X_H corresponde ao valor final do campo magnético compensado no eixo X ;
- Y_H corresponde ao valor final do campo magnético compensado no eixo Y ;
- M_x corresponde ao valor do campo magnético no eixo X previamente calibrado;
- M_y corresponde ao valor do campo magnético no eixo Y previamente calibrado;

Para obtenção dos ângulos *roll* e *pitch* é necessário utilizar um método que permita estimar corretamente estes dois ângulos. Podemos calcular esses ângulos de duas maneiras, utilizando o acelerómetro ou o giroscópio, no entanto, ambos os sensores têm suas imprecisões. Começando pelo acelerómetro, podemos estimar os ângulos *pitch* e *roll* com base nas seguintes equações [72]:

$$Pitch = \beta = \arcsin\left(\frac{a_x}{a_z}\right) \quad (33)$$

$$Roll = \gamma = \arctan\left(\frac{a_y}{\sqrt{a_x^2 + a_z^2}}\right) \quad (34)$$

onde:

- a_x, a_y e a_z correspondem aos valores de aceleração dados pelo acelerómetro;

No caso do acelerómetro, é importante notar que ele tende a gerar bastante ruído, e esse ruído pode aumentar quando o sensor está em contato com a estrutura mecânica do robô [72].

Por outro lado, o giroscópio mede diretamente a taxa de rotação. Se conhecermos o ângulo inicial do sistema e a taxa de amostragem, podemos usar as leituras do giroscópio para integrar os dados e calcular o ângulo ao longo do tempo. No entanto, é essencial observar que esse procedimento funciona bem em curtos períodos, mas em períodos mais longos, pode ocorrer um fenómeno conhecido como "*gyro drift*" [72].

Assim sendo, é necessário um método que permita a precisão a curto prazo das estimativas dos ângulos do giroscópio e a estabilidade a longo prazo das estimativas dos ângulos do acelerómetro.

Assim sendo, utiliza-se o filtro complementar, abordado na secção 2.8 que soluciona o problema em questão, fornecendo assim os ângulos *roll* e *pitch* para serem aplicados nas equações 31 e 32. Em ROS já existe uma *package* que permite aplicar o filtro complementar (estando esta *package* disponível em [73]) e desta forma ajustou-se esta *package* de forma a conseguir subscrever os tópicos dados pelo IMU do *Duro Inertial* para obter os ângulos *roll* e *pitch*.

3.7.4 Cálculo do Heading

Com base nos métodos/procedimentos abordados nas três últimas subsecções, é possível calcular o *Heading* compensado das interferências e do *tilt* através do uso do filtro complementar através das equações:

$$\text{Heading} = \text{atan2} \left(\frac{M_{yH}}{M_{xH}} \right) \text{ radians} \quad (35)$$

$$\text{Heading} = \text{atan2} \left(\frac{M_{yH}}{M_{xH}} \right) \times \frac{180}{\pi} \text{ degrees} \quad (36)$$

3.8 Aquisição da Posição

A forma como os robôs são localizados no seu espaço de trabalho pode variar de acordo com as necessidades do sistema. No caso da MDB LV600 PRO, a sua posição é determinada usando informações de GPS e com base no vetor direção dado pelo magnetómetro. Com base nesses dados, é possível representar a posição do robô num espaço bidimensional, como ilustrado na Figura 58.

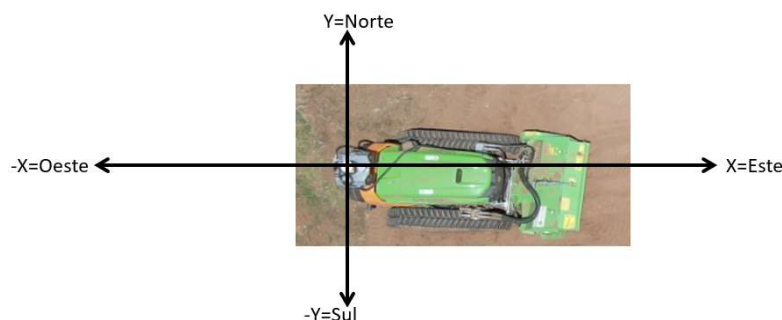


Figura 58 – Modelo de posição MDB LV600 PRO

Para ser possível a representação da figura 58, é necessário adotar um sistema de coordenadas que permita realizar a correspondência entre o norte magnético o eixo Y. Desta forma as coordenadas UTM, como abordado na secção 2.6 permitem satisfazer os requisitos do sistema. Portanto é necessário realizar a conversão dos dados fornecidos pelo *Duro Inertial*, no que diz respeito à posição, ou seja, latitude e longitude e converter para coordenadas UTM. Para tal utiliza-se uma *package* em ROS que já realiza esta conversão, denominada por “*gps_common*”, estando o

código fonte disponível em [74], que vai subscrever a mensagem dada pelo GPS do *Duro Inertial* e publicar um tópico com uma mensagem de coordenadas UTM.

Como referido na secção 3.2 o Sentry é equipado com um sistema RTK. Este sistema permite obter uma precisão na posição que pode chegar a 1 cm [75], o que é bastante aceitável para um robô com as dimensões apresentadas na tabela 1. Com base no referido a posição proveniente do *Duro Inertial* não necessita de ser submetida por filtragem devido ao RTK, no entanto este sistema necessita de ter linha de vista desobstruída para funcionar corretamente.

Na floresta, onde a vegetação é densa, árvores e obstáculos como rochas podem obstruir a comunicação direta entre o sistema da estação base e o sistema RTK do robô, assim sendo é crucial desenvolver um cenário alternativo para determinar a posição do robô quando o sistema RTK não está a funcionar. Portanto, os testes realizados e os algoritmos implementados neste estudo para determinar a posição do robô foram conduzidos excluindo a dependência do sistema RTK, logo a aplicação de filtros deve de ter em conta os erros que um GPS pode originar.

O diagrama de blocos representado na Figura 59, ilustra o conjunto de etapas necessárias para determinar a posição do robô. Neste processo, as coordenadas de latitude e longitude são primeiramente convertidas para o sistema UTM, seguido da aplicação de um filtro, resultando, por fim, nas coordenadas estimadas X e Y do robô.

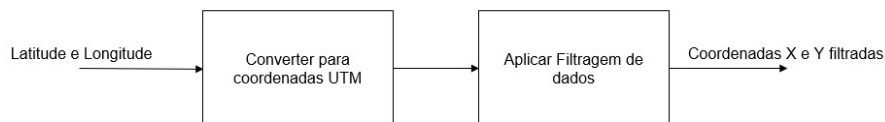


Figura 59 – Diagrama de blocos para obter posição filtrada

Para efetuar a filtragem dos dados da posição X e Y, foram analisadas duas técnicas distintas de filtragem: a utilização de um *Simple Moving Average Filter* (SMAF) e um filtro de Kalman. As próximas secções abordarão a implementação desses filtros. A aplicação destes filtros resulta de um estudo realizado por *S. Han et al.* em [46].

3.8.1 Filtro SMA

Referenciando a secção 2.7.3, e observando a Figura 32, o SMA é um dos tipos de filtros da família dos filtros de média móvel. O *Simple Moving Average* é uma média não ponderada definida por k pontos e segue o seguinte modelo de equação:

$$SMA = (x_1 + x_2 + x_3 + \dots + x_k)/k \quad (37)$$

Onde $x_1, x_2, x_3, \dots, x_k$ são os k pontos de dados consecutivos. O SMA representa a média aritmética desses k pontos de dados.

Os pontos obtidos são guardados numa lista com tamanho fixo definindo assim a variável k como o tamanho da lista.

3.8.1 Filtro de Kalman

O filtro de Kalman aplicado neste trabalho para fornecer posição, vai integrar os dados provenientes do GPS e do IMU. Os dados provenientes do IMU são utilizados para projetar a etapa de previsão onde se vai obter uma estimativa do próximo estado do sistema que neste caso é a posição. Enquanto isso, na etapa de atualização, os dados do GPS entram no filtro. Nesta fase, a posição prevista pelo IMU é comparada com as leituras reais do GPS, e a diferença entre esses valores é calculada. Essa diferença leva a um processo de correção, e é aqui que o filtro de Kalman entra em ação. Ele calcula um ganho que determina a importância relativa dos dados do IMU e do GPS na obtenção de uma estimativa de posição mais precisa. Em resumo, o filtro de Kalman equilibra as informações do IMU e do GPS para fornecer uma estimativa precisa da posição atual.

O diagrama de blocos da Figura 60 representa o sistema que envolve a integração do IMU com o GPS num filtro de Kalman.

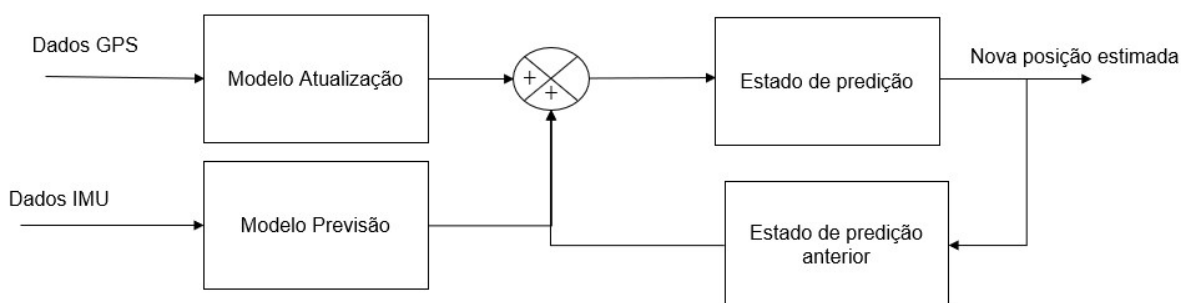


Figura 60 – Funcionamento do Filtro de Kalman onde existe a fusão entre o GPS e IMU para obter a posição

A aplicação do Filtro de Kalman partiu de uma *package* em ROS denominada por “*robot_pose_EKF*” onde o código fonte está disponível em [76]. Através desta Package manipulou-se os dados de entrada para os dados provenientes do GPS em coordenadas UTM, e do IMU. Desta forma e observando a Figura 60 a fusão destes dois sensores vai resultar numa posição combinada.

3.9 Algoritmo do Sistema de Visão

Como já indicado na secção 3.3 o Sentry tem presente um sistema de visão composto por duas câmaras RGBD e duas câmaras térmicas. O trabalho realizado

focou-se na análise dos dados provenientes das câmaras RGBD. Implementou-se um algoritmo destinado à identificação de objetos, com ênfase particular na deteção de pessoas. Além disso, o algoritmo foi utilizado para medir as distâncias até aos obstáculos no ambiente, com a finalidade de mudar o estado da máquina. O diagrama de blocos apresentado na Figura 61 mostra o funcionamento do sistema de visão incorporado no sistema Sentry.

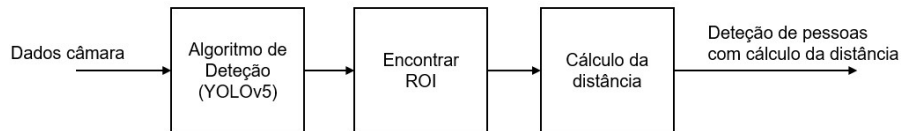


Figura 61 – Modelo do sistema de visão

O algoritmo de deteção de obstáculos usado foi o YOLOv5 como já mencionado na secção 2.9, estando o código fonte disponível em [77]. Para configurar o YOLOv5 na Jetson Xavier Nx, foi necessário proceder à instalação das seguintes bibliotecas:

- PyTorch: consiste numa biblioteca de *Machine Learning* (ML) que oferece suporte à criação e treino de rede neurais [78];
- CUDA e cuDNN: o CUDA é uma plataforma de computação da NVidia que permite utilizar a potência de processamento da GPU e desta forma quando se lança o YOLOv5 este corre pela GPU. O cuDNN é uma biblioteca de ML que acelera operações relacionada com redes neurais [79];
- OpenCV: consiste numa biblioteca usada para o processamento de imagens e visão computacional [80];
- Torchvision: É uma biblioteca que fornece *datasets* [78];
- Numpy: É uma biblioteca para computação numérica em Python [81].

O cálculo da distância é realizado por via das câmaras RGBD (Intel Realsense). As câmaras RGB vão capturar imagens de um dado ambiente, enquanto a câmara *Depth* mede a distância dos objetos utilizando o método ToF. Para realizar as medições de distâncias recorreu-se à biblioteca *pyrealsense2* que está associada as câmaras RGBD da Intel cujo código fonte está disponível em [82].

Desta forma é necessário desenvolver um algoritmo que permita conectar a deteção de objetos por parte do YOLOv5 com a medição da distância até eles utilizando a *pyrealsense2*. Esse algoritmo pode ser observado no fluxograma da Figura 62.

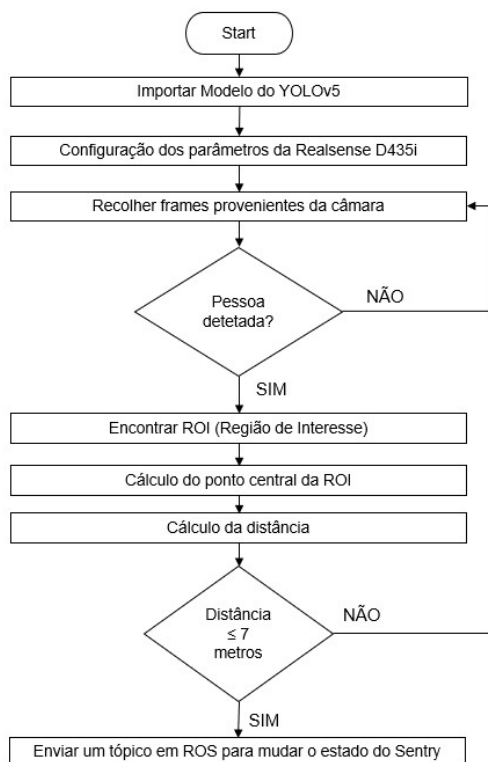


Figura 62 – Processo para detecção de pessoas com distância usando o YOLOv5

O processo começa com a importação do modelo treinado pelo YOLOv5, seguido pela configuração dos parâmetros da câmara Intel Realsense. Uma vez concluída essa etapa, a câmara adquire imagens. Se forem detetadas pessoas com uma taxa de confiabilidade de 70% (sendo esta taxa fornecida pelo YOLOv5 cada vez que deteta um objeto), inicia-se um processo de identificação da localização usando a região de interesse. Isso envolve encontrar o centro dessa área específica. Com base nesse ponto central, a medição de distância é realizada. Para o caso deste trabalho, foi estabelecido que, se a distância for menor ou igual a 7 metros, o sistema aciona uma paragem de emergência. Isso significa que um sinal booleano é enviado para o sistema do Sentry, que altera o estado da máquina e a coloca num estado de paragem de segurança. Este sinal Booleano necessita de ser um tópico em ROS, para que quando a máquina esteja futuramente em navegação autónoma, alterar o seu estado imediatamente.

3.10 Interação com o Microcontrolador

Como mencionado na seção 3.2, o microcontrolador Arduino Portenta vai servir como um controlador de assistência à Nvidia Jetson Xavier Nx. A sua principal finalidade vai servir para controlar os movimentos mecânicos da Máquina Florestal. Desta forma, por meio do ROS, utilizando a Package “*Rosserial*”, permite a comunicação entre a Jetson e o Arduino.

Para documentar essa interação é necessário apresentar primeiro os conceitos gerais subjacentes ao controlo de todas as ações da Máquina LV600 PRO, de

seguida é descrito o princípio de funcionamento que está por detrás do sistema de locomoção da referida Máquina.

3.10.1 Controlo da Máquina LV600 PRO

O controlo da Máquina LV600 PRO é realizado por sinais digitais e analógicos. Os sinais digitais são auxiliados por relés que enviam a tensão desejada para controlar certas funcionalidades da máquina como aquecimento de resistência, ignição, controlo dos movimentos da ferramenta, controlo do destroçador e ativação do movimento das lagartas. Os sinais analógicos estão divididos em duas partes sendo uma a parte das rotações do motor a combustão, que são controladas por sinais analógicos enviados pelo Arduino. O sistema de locomoção da máquina é composto por Drivers, cuja integração será explicada próxima secção. O envio destes sinais para cada pino da ficha de 64 pinos irá originar numa ação/movimento no trator.

No entanto para trabalhar com a máquina é necessário um conjunto de procedimentos que são realizados de forma hierárquica para que seja possível movimentar a máquina durante a navegação autónomo. A título de exemplo a Figura 63, representa a máquina de estados empregue para ligar o trator. No estado A é realizado o aquecimento da resistência do trator durante 6 segundos, e após este tempo o estado B liga a ignição durante 4 segundos. Desta forma resulta no funcionamento do motor a combustão. O estado C controla as rotações do motor, sendo que estas necessitam de ser aumentadas gradualmente e quando estas atingirem um valor predefinido a máquina de estados avança para o estado D que ativa as lagartas para que seja possível o movimento destas, durante a navegação. No entanto, caso seja premido o botão de emergência a máquina de estados salta logo para o estado E. Esta máquina de estados resulta de um conjunto de testes realizados, onde se realizou a medição dos sinais de cada pino da ficha 64 pinos e retirou-se as conclusões que efetivam o funcionamento da máquina respeitando questões de segurança do trator e do motor de combustão.

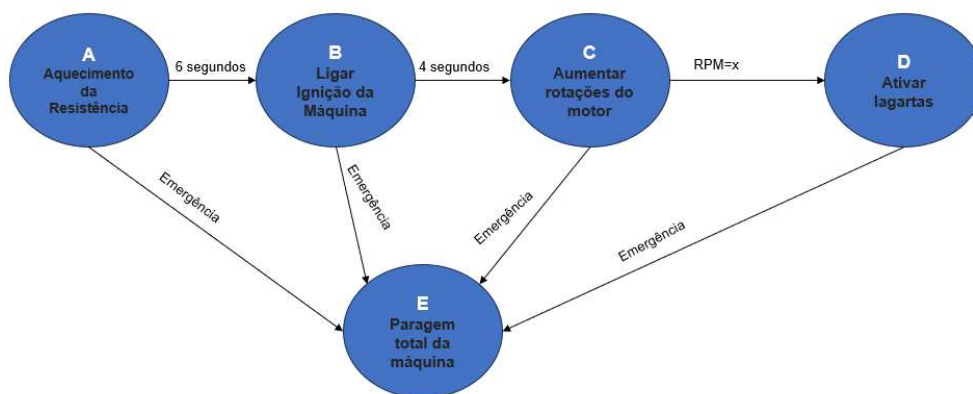


Figura 63 – Máquina de estados para o controlo da LV600 PRO

3.10.1 Sistema de locomoção

O sistema de locomoção do robô segue a arquitetura definida na Figura 43. Neste processo, há uma interação entre o ROS e o Arduino, onde este após receber um valor proveniente do ROS converte-o num sinal de tensão que corresponde a uma determinada velocidade da máquina LV600 PRO. O ROS transmite uma mensagem que o microcontrolador recebe e interpreta, resultando num valor decimal no intervalo de 0 a 255. Esse valor é posteriormente convertido numa tensão na gama de 0 a 5 V. Entretanto, como explicado anteriormente, o microcontrolador não possui capacidade para fornecer o nível de energia adequado às electroválvulas responsáveis pelos movimentos do trator. Portanto, o *driver* DRI0041 entra em ação, ajustando a tensão inicialmente enviada pelo microcontrolador para fornecer os valores de tensão e corrente necessários para movimentar as lagartas do trator. Para o efeito, foram estabelecidos limites superiores e inferiores para a rotação pura e o deslocamento linear, com base num teste realizado ao operar a máquina por meio do sistema de teleoperação.

Esses valores são ajustados conforme necessário, variando de 0 a 255 em formato decimal, dependendo das condições do terreno e da velocidade desejada. A tabela 4 exhibe os valores escolhidos para as velocidades lineares e angulares do robô.

Tabela 4 – Valores definidos no Microcontrolador

	Linear (decimal)	Angular (decimal)
Mínimo	70	85
Máximo	110	110

Com base nos dados da Tabela 4, estabelecemos uma relação usando interpolação linear para mapear os valores provenientes da mensagem do ROS. Os valores mínimo e máximo da mensagem do ROS variam de 0.1 a 1.0, tanto para a velocidade linear quanto para a velocidade angular. Para tal foram exploradas três abordagens diferentes para o sistema de locomoção. A primeira abordagem envolve uma solução direta, na qual os movimentos lineares e angulares são tratados separadamente. A segunda adota uma função linear que combina esses dois tipos de movimentos, enquanto a terceira opta por uma função não linear para aprimorar o controlo das lagartas.

A primeira abordagem é uma solução simplificada que permite que o robô se mova em dois estados distintos: movimento linear e movimento angular, mas esses movimentos são descontínuos. Para o movimento de translação retilínea, como ilustrado na Figura 64 (a), ambas as lagartas se movem para a frente com a mesma velocidade. O movimento de rotação, conforme ilustrado na Figura 64 (b), representa uma conceção do comportamento real da máquina, onde ela gira sobre seu próprio eixo. Nesse movimento, as duas lagartas giram em direções opostas com a mesma velocidade.

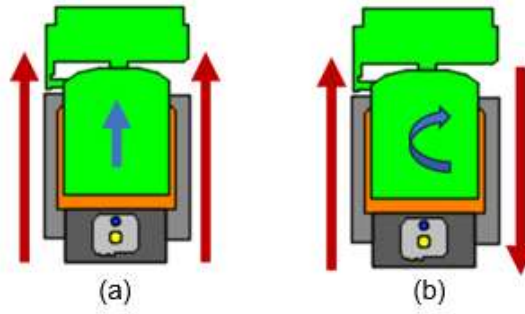


Figura 64 – Método locomoção primeira abordagem. (a): locomoção do trator numa trajetória retilínea; (b): Movimento rotacional do trator no seu próprio eixo

A segunda abordagem, representada nas figuras 65 (a) e 65 (b), evolui a partir da abordagem anterior ao permitir a combinação dos movimentos linear e angular. No entanto, requer a definição de valores de tensão distintos para que as electroválvulas recebam o mesmo sinal. A equação que regula a transformação é a seguinte:

$$x = \frac{Max_{lin_{level}} - Min_{lin_{vel}}}{0.9} \times actual_{lin_{value}} + Max_{lin_{vel}} - \frac{Max_{lin_{level}} - Min_{lin_{vel}}}{0.9} \quad (38)$$

Desta forma é possível regular a velocidade de cada lagarta utilizando a seguinte equação:

Lagarta esquerda com mais potência que a direita:

$$Vel_{right_{whell}} = x \times (1 + actual_{angle_{value}}) \quad (39)$$

$$Vel_{left_{whell}} = x \quad (40)$$

Lagarta direita com mais potência que a esquerda:

$$Vel_{right_{whell}} = x \quad (41)$$

$$Vel_{left_{whell}} = x \times (1 + actual_{angle_{value}}) \quad (42)$$

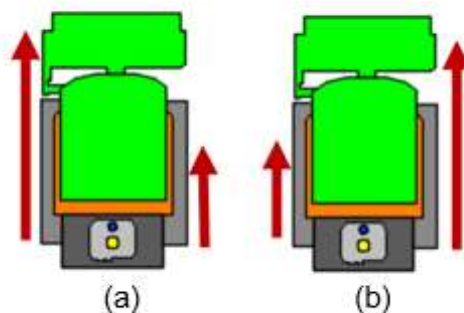


Figura 65 – Movimentos combinados. (a): Movimento combinado onde a lagarta esquerda tem mais força que a direita; (b): Movimento combinado onde a lagarta direita tem mais força que a esquerda

A terceira abordagem introduz uma função não linear com o propósito de aprimorar a abordagem anterior. Essencialmente, o sistema é discretizado de modo a tornar o movimento das lagartas o mais suave possível, mesmo em velocidades mais baixas, o que facilita o processo de navegação.

3.11 Detecção de Obstáculos com o LIDAR

De forma a permitir que o robô se desvie de obstáculos com sucesso, é essencial que ele seja capaz de identificar o que está dentro do seu FOV. Nesse sentido, foi incorporado um algoritmo capaz de detectar a presença de objetos, tanto estáticos quanto dinâmicos. Além disso, é crucial determinar as dimensões desses obstáculos. Para alcançar esse objetivo, usou-se o algoritmo *adptive_clustering*, desenvolvido com base no estudo [83] e com o código fonte disponibilizado em [84] para interação entre a *point cloud* do LIDAR VLP-16 com o ROS.

O processo de agrupamento (*clustering*) dos dados da varredura do LIDAR desempenha um papel fundamental na detecção de objetos em ambientes específicos. Com a ajuda do agrupamento, podemos identificar e distinguir diferentes objetos presentes. Entretanto, no âmbito deste projeto, a função principal do *clustering* do LIDAR é detectar a presença de objetos, sem a necessidade de uma análise mais aprofundada. O *clustering* traz consigo uma vantagem significativa, uma vez que reduz consideravelmente a carga de processamento dos dados provenientes da nuvem de pontos. Isso é alcançado ao definir regiões de interesse e distâncias específicas, o que, por sua vez, simplifica o processamento dos dados obtidos pelo LIDAR. O agrupamento opera com base numa entrada que representa uma nuvem de pontos 3D, conforme definido da seguinte maneira [83]:

$$P = \{p_i | p_i = \{x_i, y_i, z_i\} \in R^3, i = 1, \dots, I\}$$

Onde I corresponde ao número total de pontos num *scan*. Para conseguir eliminar pontos pertencentes ao chão, este algoritmo determina uma distância Z_i que tem de ser inferior a um Z_{min} [44]. Este algoritmo opera com uma distância, definida como ' d ', que determina o limite pelo qual o *clustering* vai ser feito nos objetos em redor num ambiente. Se esse limite for muito pequeno, pode resultar na divisão de objetos individuais em múltiplos *clusters*, enquanto um limite muito grande pode levar à integração de vários objetos num único *cluster*. Além disso, uma distância maior requer mais poder computacional para processamento e formação de *clusters* [83].

Quando o algoritmo "*adaptive_clustering*" está ativo, ele envia uma mensagem por meio do ROS contendo informações sobre os obstáculos, consolidando todos os detalhes de cada *cluster* identificado. Visualmente, cada *cluster* é representado por um polígono, que inclui as coordenadas máximas e mínimas de X e Y dos pontos que o compõem. Essas coordenadas são cruciais para determinar a distância em relação ao LIDAR, no caso específico, o Velodyne, uma vez que são perpendiculares a ele. Em seguida, o algoritmo projeta as coordenadas da caixa de cada *frame* no mapa, considerando a posição e orientação do robô. Como resultado, o algoritmo fornece as coordenadas dos quatro vértices da caixa, permitindo a obtenção dos valores máximos e mínimos de X e Y em cada polígono.

3.12 Arquitetura de Navegação

O objetivo principal deste trabalho é garantir a confiabilidade dos dados, permitindo que, quando o algoritmo de navegação for aplicado, a máquina florestal possa seguir trajetórias de maneira autónoma e segura. A Figura 66 fornece uma visão completa da arquitetura do sistema Sentry, que opera em conjunto com todos os sensores. Esses sensores estão integrados ao ROS e fornecem tópicos contendo dados que precisam ser processados e filtrados. Após essa manipulação dos dados, eles são publicados no ROS, onde podem ser subscritos, fortalecendo assim a robustez do algoritmo de navegação.

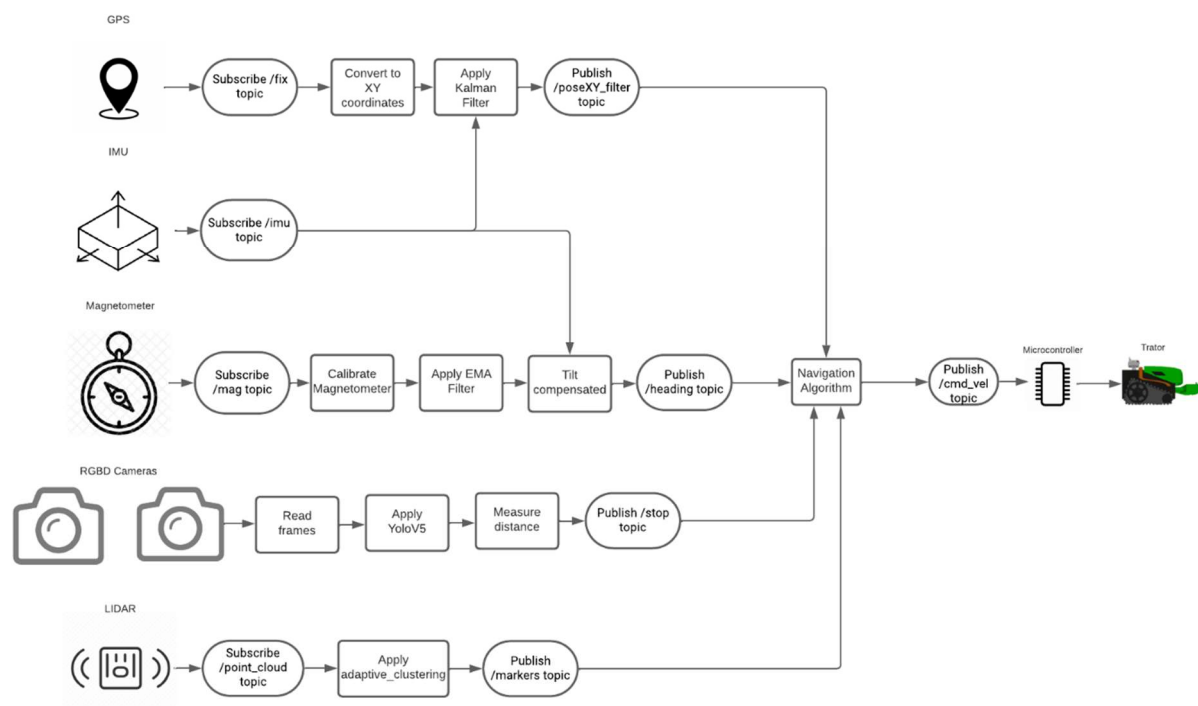


Figura 66 – Arquitetura Final do Sistema Sentry para incorporar Algoritmos de Navegação

3.12 Funcionamento Prático do Sistema Sentry

Como referido anteriormente a integração de um router 3G/4G serve para que o Sentry comunique com uma plataforma web. Esta plataforma ainda se encontra em desenvolvimento por parte de outra entidade do projeto F4F, onde o objetivo é proporcionar ao utilizador o controlo sobre a máquina.

O princípio funcionamento da plataforma web em cooperação com o Sentry, vai permitir ao utilizador, a partir de um mapa, selecionar áreas de interesse para o corte de vegetação numa determinada área o robô vai-se localizar, e para além de selecionar estas áreas, também pode selecionar locais para os quais o robô não se pode deslocar. Desta forma o robô deve autonomamente dirigir-se para as áreas de interesse e realizar o desbaste da vegetação. O processo de desbaste é realizado como um cenário real da máquina, onde esta se desloca 2 metros para a frente a desbastar e 1 metro para trás para alisar a vegetação cortada.

Esta plataforma também serve para o utilizador verificar o estado da máquina, como a visualização do que as câmaras estão a ver, ou eventuais erros que possam interditar de certa forma o ciclo de trabalho da máquina e desta forma é feita a gestão pelo utilizador de forma a gerir todos os processos.

CAPÍTULO 4: Discussão de Resultados

Os testes práticos realizados pretenderam testar e validar todos os algoritmos e funcionalidades implementados nos sensores do Sentry. Primeiramente foram realizados testes à máquina através do sistema de teleoperação, de modo que esta fosse controlada por um operador, efetuou-se o teste da orientação e da posição do robô, testou-se o algoritmo de deteção através do sistema de visão validando o algoritmo *adaptive clustering* referente ao LIDAR.

As experiências realizadas foram efetuadas em terreno anexo ao Departamento de Engenharia Mecânica da FCTUC conforme se mostra na Figura 67.



Figura 67 – Local de testes experimentais

4.1 Teleoperação

Este teste teve como objetivo verificar se os comandos enviados pelo sistema ROS resultam nos movimentos desejados da LV600 PRO. A Figura 68 apresenta um esquema do procedimento utilizado para realizar a teleoperação da máquina utilizando o sistema Sentry. Isto permitiu o controlo completo das ações da máquina, e para alcançar esse controlo, validando o funcionamento da Package “*rosserial*”.

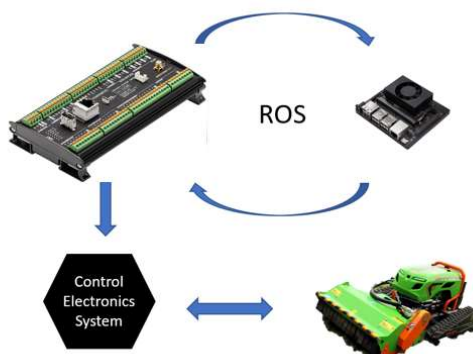


Figura 68 – Interação entre o ROS e a MDB LV600 PRO

4.2 Teste da Orientação do Robô

A avaliação prática da orientação do robô segue o mesmo procedimento indicado no diagrama de blocos da Figura 50. Neste teste, avaliamos o impacto da calibração do sensor, da aplicação de filtros nos dados e da correção do *tilt*.

4.2.1 Desempenho da Calibração

Conforme demonstrado no fluxograma da Figura 52, o processo de calibração do magnetómetro foi realizado, envolvendo a recolha de 1000 pontos em cada eixo do magnetómetro. Em seguida, calculou-se os valores de *Bias* e do Fator de Escala. A Figura 69 ilustra claramente o impacto da calibração do magnetómetro, permitindo a comparação entre dados do sensor calibrado e um não calibrado. Esse método ajusta os dados de uma circunferência para o centro de um referencial, como demonstrado na Figura 69.

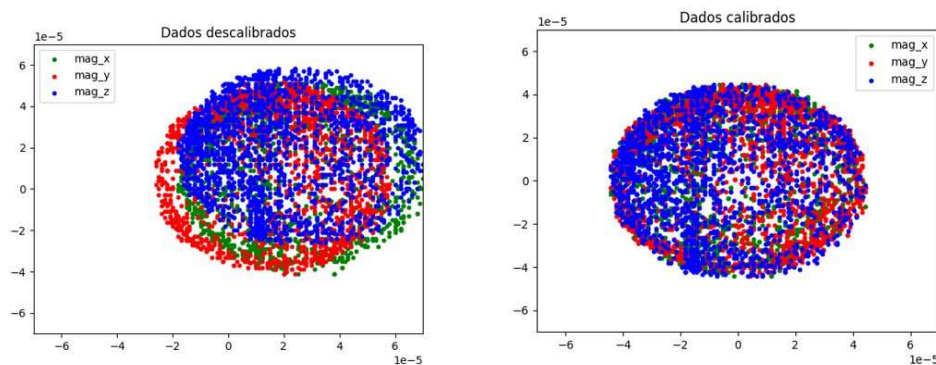


Figura 69 – Comparação dados calibrados vs dados descalibrados

No exemplo seguinte, procedeu-se ao teste para calcular o *heading*, que envolve verificar se as direções Norte e Este magnéticas do sensor correspondem ao valor correto do *heading*, variando entre 0° e 90° , respetivamente. Para realizar esse teste, foi utilizado um dispositivo Pixhawk px4, que é um equipamento munido com vários sensores, incluindo GPS, IMU, magnetómetro, entre outros. Ele fornece informações precisas sobre as direções Norte e Este magnético, bem como as direções Norte e Este da Terra. Após a calibração do magnetómetro do *Duro Inercial*, posicionou-se o dispositivo em paralelo com o equipamento Pixhawk px4, alinhando-os na mesma direção e sentido, conforme mostrado na Figura 70.

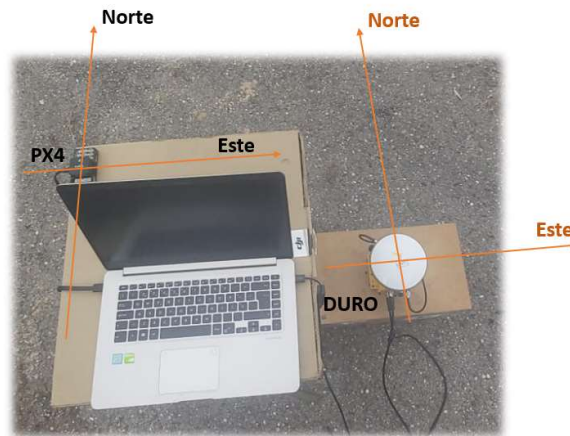


Figura 70 – Pixhawk Px4 e Duro Inertial em paralelo para aferir o Heading

Esta comparação indicou um *offset* de 86° entre o Norte Magnético do Duro e o Norte magnético do Pixhawk. Esta discrepância ocorre devido à orientação específica do Magnetômetro BMM150 dentro do *Duro Inertial*.

No entanto, é importante destacar que esse desvio pode ser prontamente corrigido aplicando uma matriz de rotação de 86° no eixo Z (utilizando a equação 3). Desta forma ajustará a orientação do eixo vertical para coincidir com a direção correta do Norte e Este magnético. Após a aplicação da matriz de rotação em Z, consegue-se alinhar com sucesso as direções do Norte e Este magnético, como ilustrado na Figura 71, comprovando o correto funcionamento da calibração do magnetômetro.

Para além disso, estão representadas várias rotações em torno do eixo Z onde se verificou que o Norte, Sul, Este e Oeste magnético, coincidiu com a mesma da *frame* do Pixhawk.

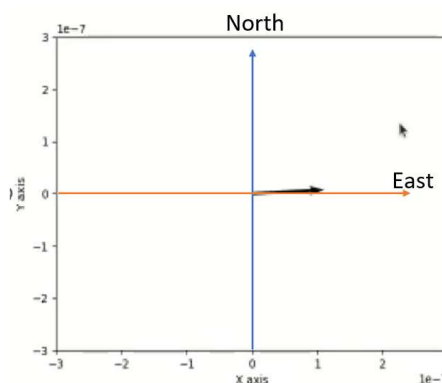


Figura 71 – Teste do Este magnético com o Eixo X da frame do Duro.

4.2.2 Desempenho da Filtragem

Para avaliar a eficiência do filtro a ser utilizado, seja ele um filtro passa-baixo LPF ou um filtro EMA, foram realizados dois testes para cada um deles. O primeiro teste ocorreu em condições estacionárias, enquanto o segundo num processo de transição.

No primeiro teste, foram calculados os seguintes parâmetros: o Máximo Erro Absoluto; a Média do Erro Absoluto e a Variância. O primeiro teste teve como objetivo manter o magnetómetro num estado de repouso, ou seja, sem qualquer vibração ou rotação, registando os valores ao longo de um período de 0,3 segundos.

No segundo teste, realizou-se a aquisição de dados durante um estado de transição. Neste teste foi necessário girar o magnetómetro numa superfície plana, abrangendo aproximadamente 120 °, a fim de observar a resposta a um impulso. Estas rotações foram realizadas manualmente, e cada rotação levou cerca de 1 segundo para ser concluída.

O objetivo deste teste consiste em determinar para o filtro passa-baixo LPF qual ordem e frequência de corte que produzem uma resposta mais rápida para os filtros, e qual o tamanho de *buffer* que se ajustava melhor às mudanças de estado para o filtro EMAF. Todos os testes foram conduzidos em ambiente laboratorial, onde o *Duro Inertial* estava posicionado numa superfície plana.

i. Testes para o filtro passa baixo LPF

Foram realizados testes com filtros passa-baixo de diferentes ordens, nomeadamente 2, 4, 6 e 8, e em cada ordem, foram adotadas diferentes frequências de corte. Tendo em conta que o valor da frequência de amostragem é igual a 25 Hz, a frequência de corte foi configurada para ter um valor inferior a 12.5 Hz (frequência de Nyquist). Para cada ordem, foram testados os seguintes valores para a frequências de corte de 2, 4, 6, 8 e 10 Hz.

Efetuuou-se o teste em regime estacionário, onde se obteve os resultados para o Máximo Erro Absoluto, a Média do Erro Absoluto e a Variância, respetivamente, para as diferentes ordens e valores de frequência de corte. Os valores destes resultados encontram-se no Anexo C.

As figuras 72, 73 e 74 mostram a média dos três eixos do magnetómetro, num gráfico de barras referente aos valores obtidos do Máximo Erro Absoluto, a Média do Erro Absoluto e a Variância, respetivamente para cada ordem.

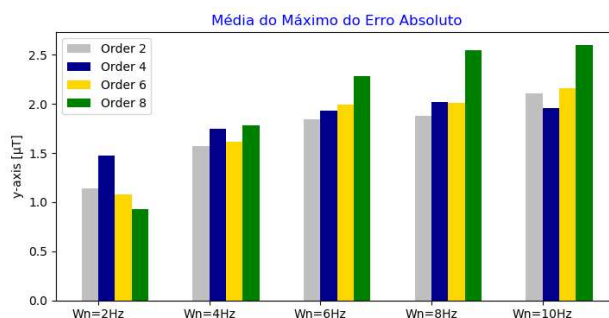


Figura 72 – Média do Máximo do Erro Absoluto no teste em regime estacionário com o LPF

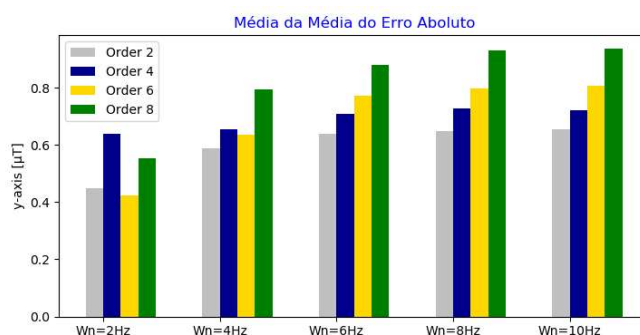


Figura 73 – Média da Média do Erro Absoluto resultante do teste em regime estacionário com o LPF

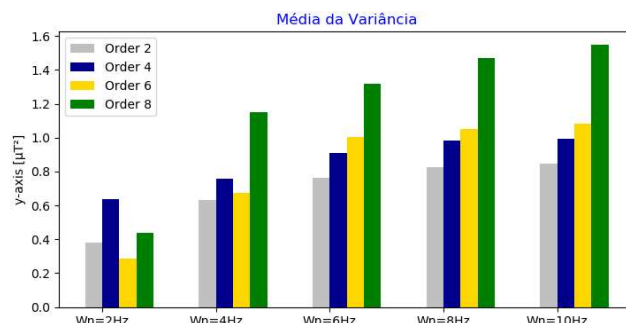


Figura 74 – Média da Variância resultante do teste em regime estacionário com o LPF

As figuras 75, 76, 77 apresentam graficamente um exemplo dos valores obtidos em regime estacionário para cada eixo do magnetômetro, X, Y e Z respectivamente e na Figura 78 mostra-se os valores obtidos referente ao *Heading*. Os gráficos apresentam valores obtidos com o filtro para diferentes frequências de corte de 2 Hz a 10 Hz e com ordem igual a 2 e os dados do magnetômetro sem aplicação de filtro. Este mesmo procedimento foi efetuado para todas as ordens mencionadas sendo estas 4, 6 e 8, e desta forma encontram-se os resultados no Anexo D.

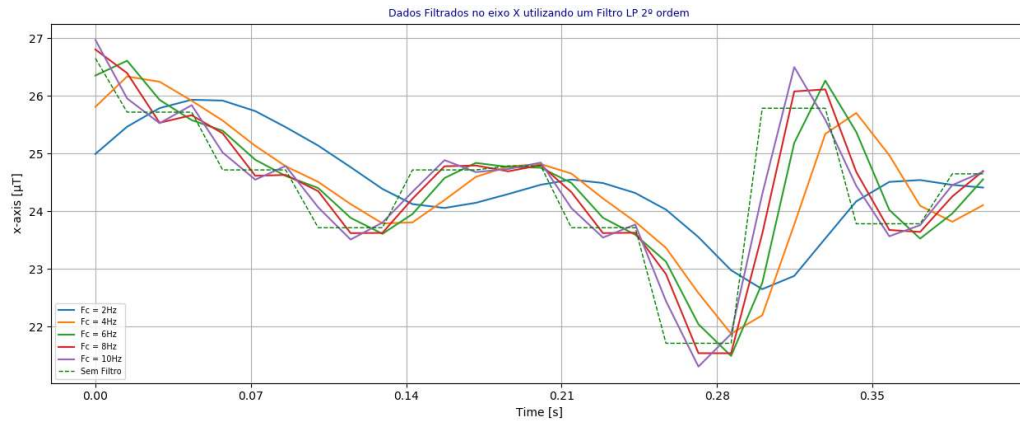


Figura 75 – Resposta temporal do LP Filter (ordem 2) referente ao eixo X

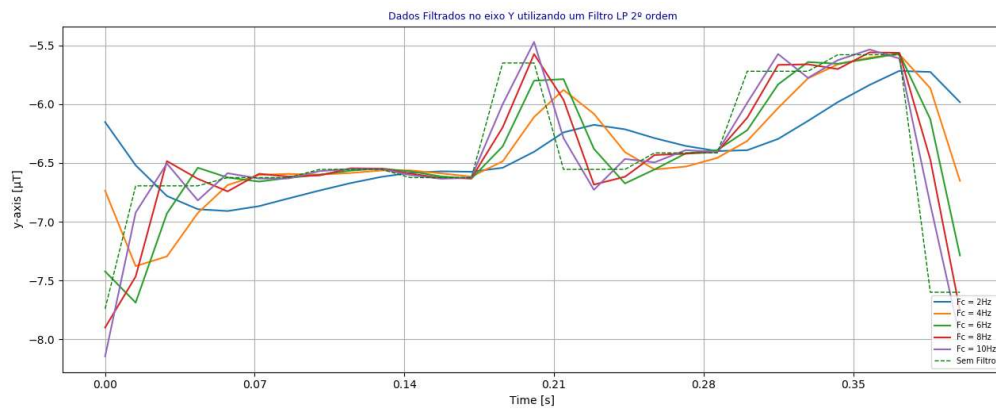


Figura 76 – Resposta temporal do LPF (ordem 2) referente ao eixo Y

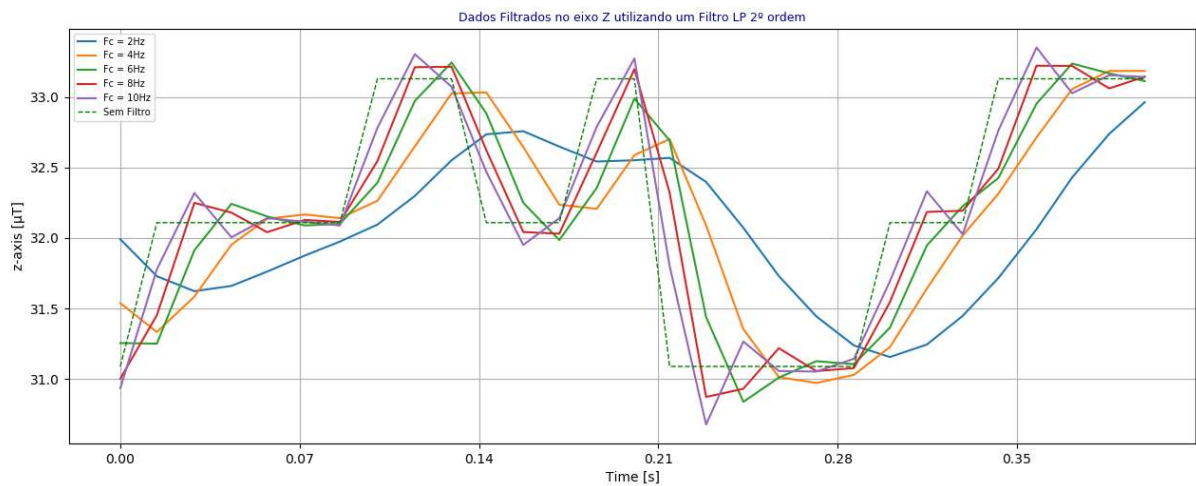


Figura 77 – Dados Filtrados em regime estacionário com o LPF (ordem 2) referente ao eixo Z

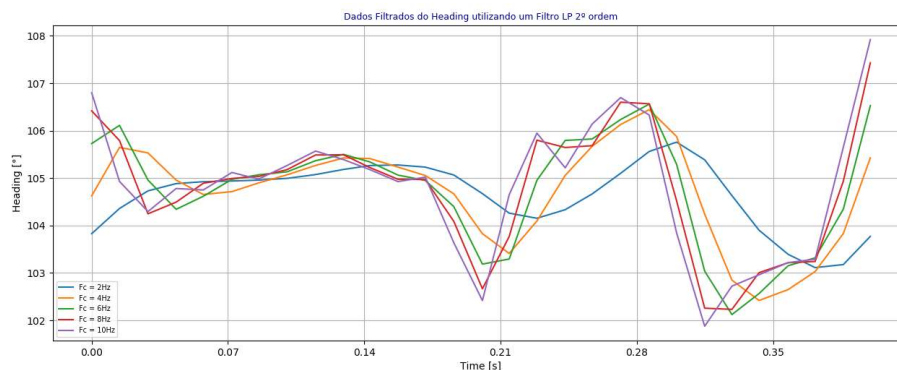


Figura 78 – Dados Filtrados em regime estacionário com o LPF (ordem 2) referente ao Heading

Na experiência para o estado de transição foi feito o processo de aquisição de dados quando se rodou o magnetômetro. Desta forma as figuras 79, 80 e 81 mostram graficamente os dados obtidos no estado de transição para o eixo X, Y e Z respectivamente e na Figura 82 mostra-se graficamente os dados do *heading*. Neste caso mostra-se a experiência realizada para um filtro passa-baixo LPF de 2ª ordem sendo que o mesmo processo foi executado para as outras ordens mencionadas, encontrando-se estas no Anexo E.

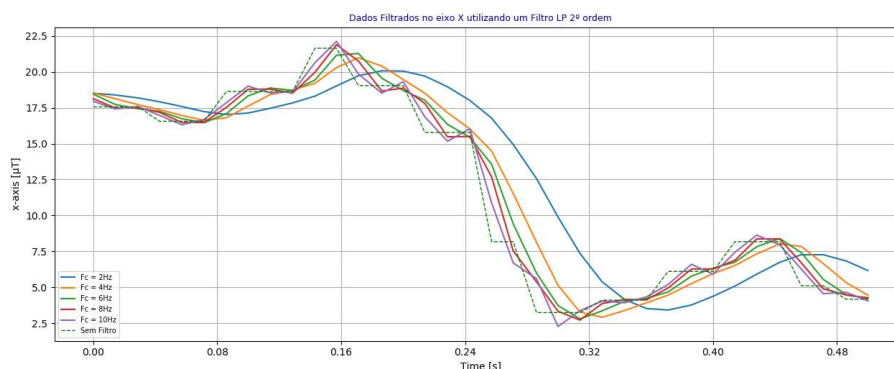


Figura 79 – Resposta temporal em regime de transição do LPF (ordem 2) referente ao eixo X

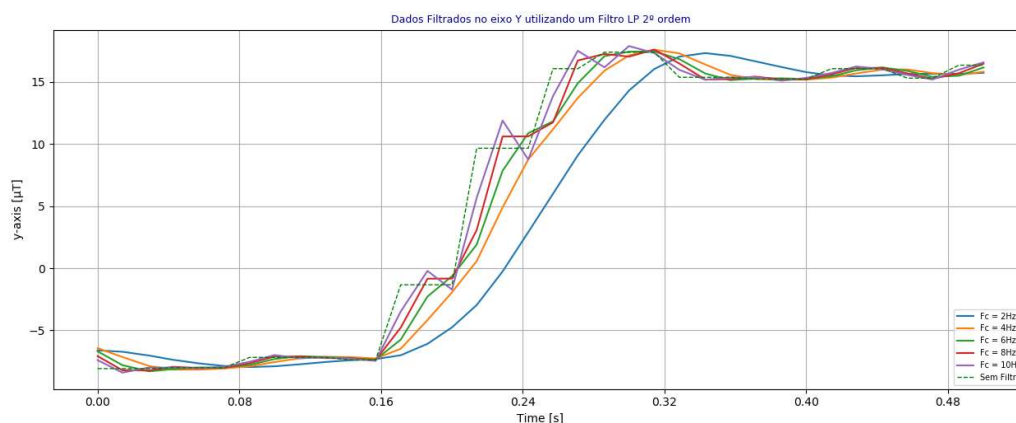


Figura 80 – Resposta temporal em regime de transição do LPF (ordem 2) referente ao eixo Y

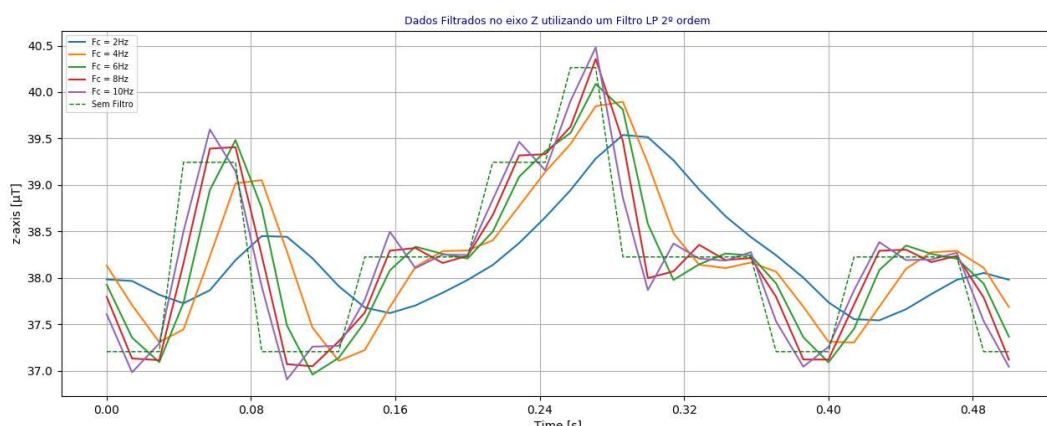


Figura 81 – Resposta temporal em regime de transição do LPF (ordem 2) referente ao eixo Z

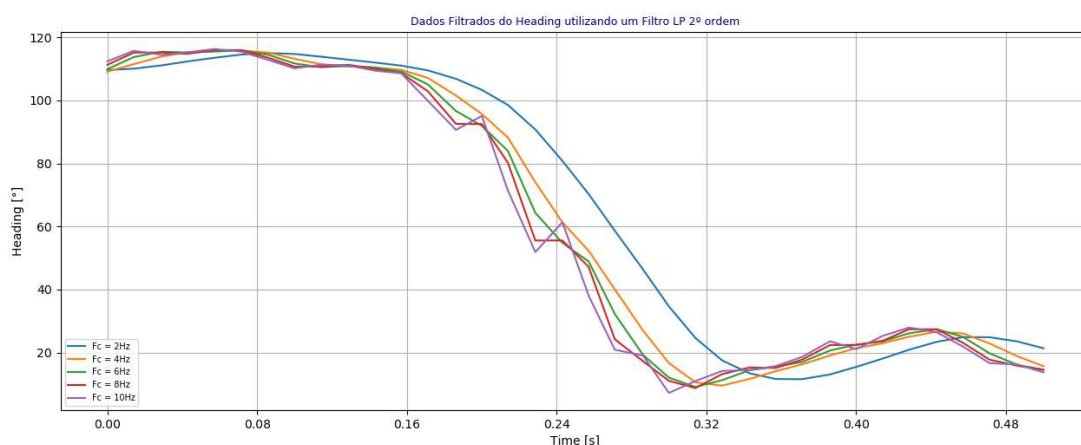


Figura 82 – Resposta temporal em regime de transição do LPF (ordem 2) referente ao Heading

Através dos dados obtidos nas experiências realizadas na transição de estado e em regime estacionário para o sistema de filtragem LPF dos dados do magnetómetro, os parâmetros que melhor se ajustam é o filtro de ordem 2 com uma frequência de corte de 4 Hz.

ii. Testes com Exponential Moving Average Filter

Para efetuar testes ao algoritmo do EMAF, foram feitos testes com diferentes tamanhos de *buffer* de dados. Os tamanhos seleccionados foram 5, 10, 15, 20 e 25. Considerou-se um teste em regime estacionário, onde se calculou os parâmetros mencionados. No Anexo F estão presentes os resultados obtidos relativos ao Máximo Erro Absoluto, Média do Erro Absoluto e Variância. As figuras 83, 84 e 85 mostram um gráfico de barras comparativo entre os vários tamanhos de um *deque* para a Média do Máximo Erro Absoluto, Média da Média do Média Erro Absoluto e Variância respetivamente.

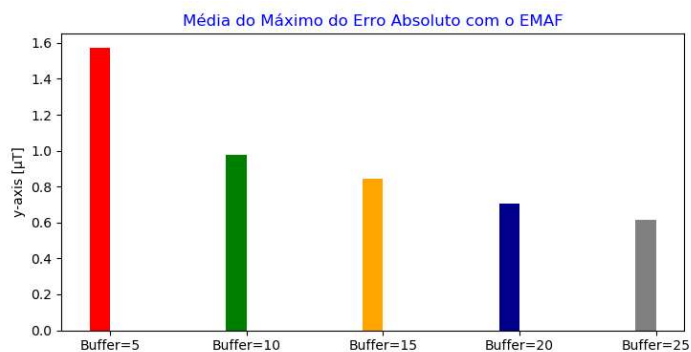


Figura 83 – Média do Máximo do Erro Absoluto resultante do teste em regime estacionário com o EMAF

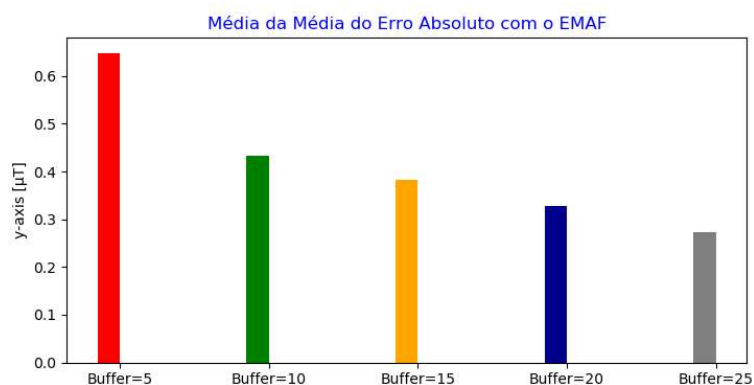


Figura 84 – Média da Média do Erro Absoluto resultante do teste em regime estacionário com o EMAF

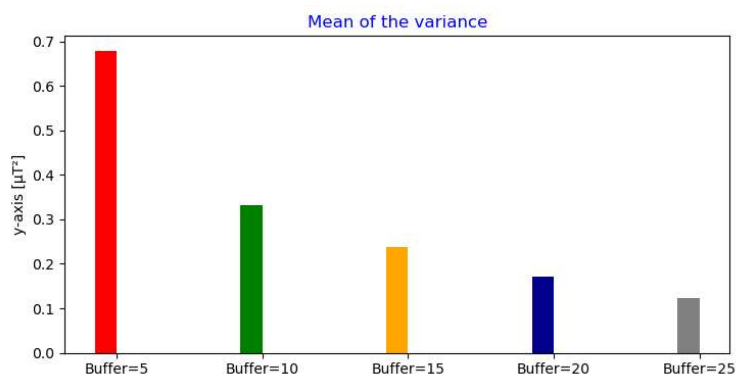


Figura 85 – Média da Variância resultante do teste em regime estacionário com o EMAF

Nas figuras 86, 87, 88 e 89 apresentam-se graficamente os valores obtidos pelo EMA para os diferentes tamanhos de *buffer* nos eixos *X*, *Y* e *Z* e no *heading* respetivamente, para um *buffer* de 5 a 25 e por último a tracejado os dados do magnetómetro sem filtro.

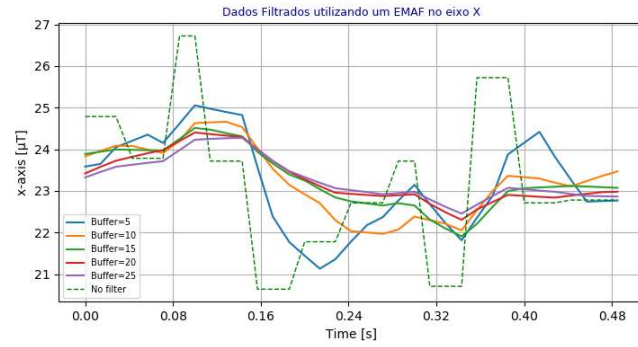


Figura 86 – Resposta temporal do EMAF referente ao eixo X em regime estacionário

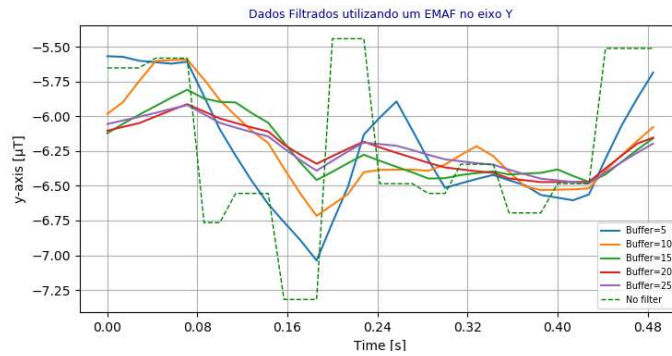


Figura 87 – Resposta temporal do EMAF referente ao eixo Y em regime estacionário

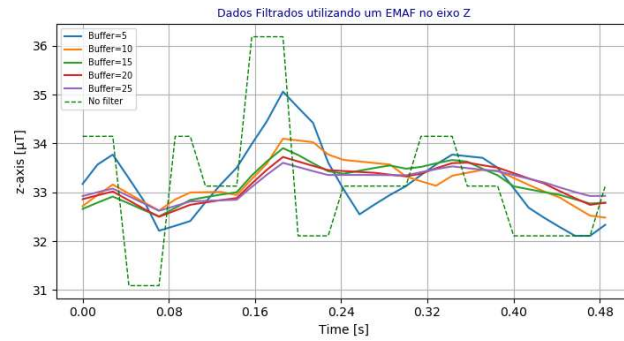


Figura 88 – Resposta temporal do EMAF referente ao eixo Z em regime estacionário

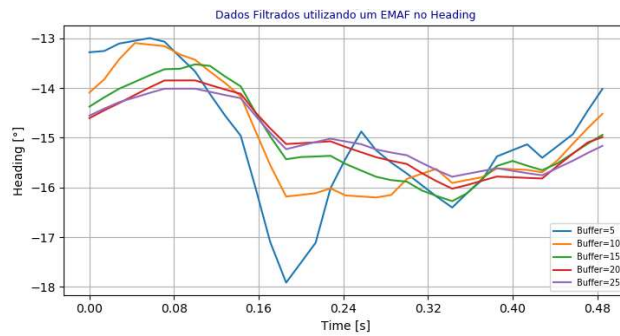


Figura 89 – Resposta temporal do EMAF referente ao Heading em regime estacionário

Da Figura 86 à Figura 89, é possível verificar que quanto maior é o *buffer* mais estabilidade vai oferecer para a filtragem dos dados do magnetômetro. No entanto quanto maior for o *buffer*, maior esforço computacional vai exigir, e por exemplo no estado de regime transitório o sistema pode ficar lento e não acompanhar o estado de rotação da máquina em tempo real. Repetiu-se a experiência, mas analisou-se o regime transitório onde se mostram os resultados nas figuras 90 a 93, dados filtrados em cada um dos eixos do magnetômetro mais o *heading*.

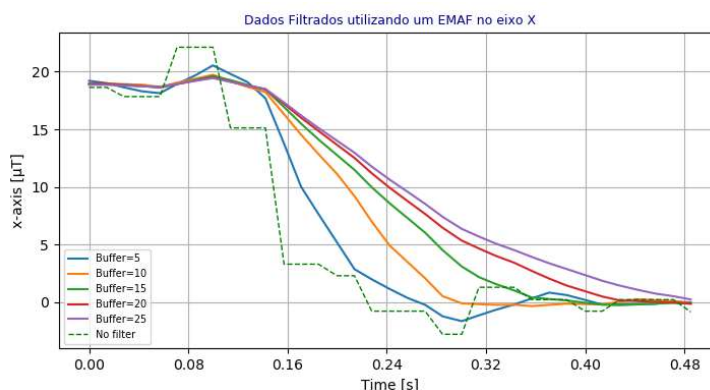


Figura 90 – Resposta temporal em regime de transição do EMAF referente ao eixo X

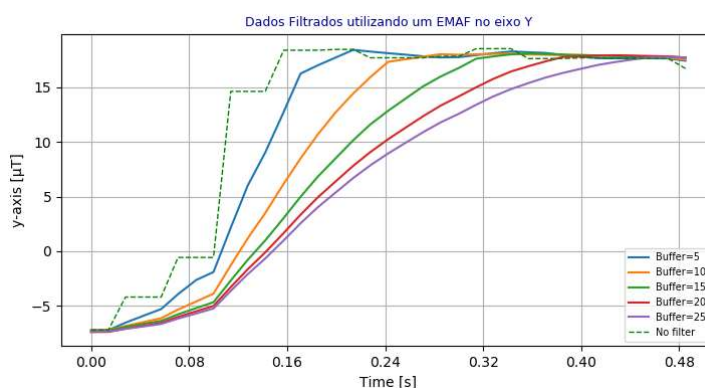


Figura 91 – Resposta temporal em regime de transição do EMAF referente ao eixo Y

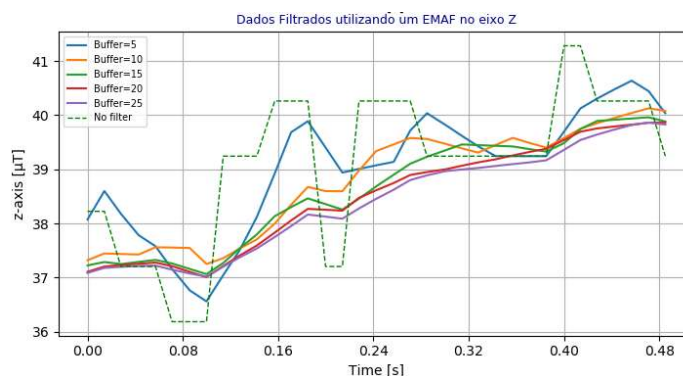


Figura 92 – Resposta temporal em regime de transição do EMAF referente ao eixo Z

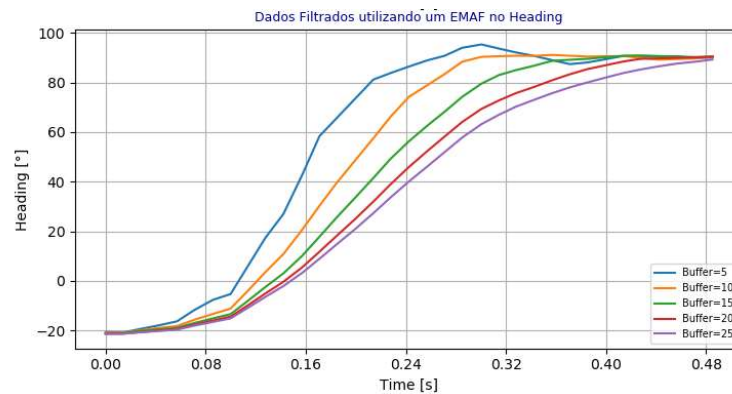


Figura 93 – Resposta temporal em regime de transição do EMAF referente ao Heading

De acordo com os dados obtidos das experiências efetuadas quer em regime estacionário quer em regime de transição, o tamanho de *buffer* que mais se adequa para filtragem no magnetómetro é um *buffer* com 10 elementos.

iii. Filtro Passa-Baixo vs *Exponential Moving Average Filter*

Após determinar os parâmetros ideais para ambos os filtros, procedeu-se a um teste comparativo para avaliar o desempenho de cada um deles. Foram realizadas as experiências em regime estacionário e de transição, utilizando o filtro passa-baixo de ordem 2 com uma frequência de corte de 4 Hz e um *buffer* de 10 elementos para o cálculo do filtro média móvel exponencial (EMAF). Os resultados das experiências em regime estacionário são apresentados nas figuras 94 a 96, mostrando um gráfico de barras comparando os parâmetros mencionados. Os valores obtidos nesta experiência encontram-se no Anexo F.

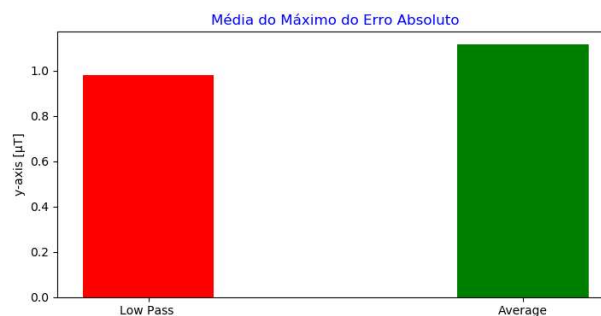


Figura 94 – Média do Máximo do Erro Absoluto resultante do teste em regime estacionário com o LPF e com o EMAF

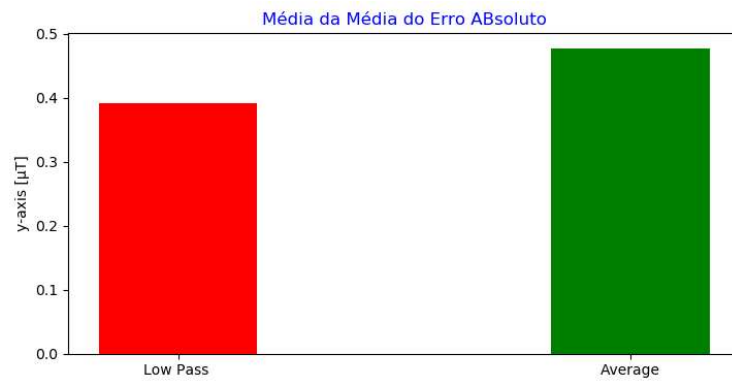


Figura 95 – Média da Média do Erro Absoluto resultante do teste em regime estacionário com o LPF e com o EMAF

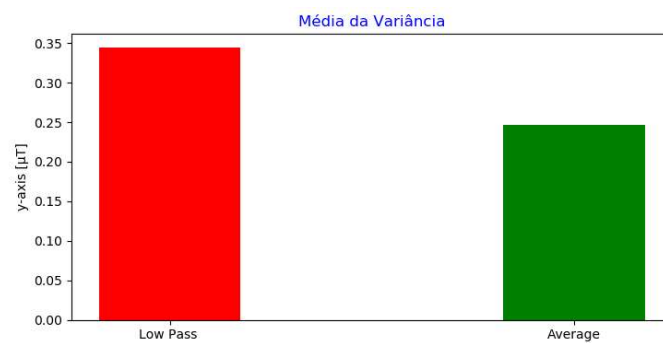


Figura 96 – Média da Variância resultante do teste em regime estacionário com o LPF e com o EMAF

O comportamento em regime estacionário de ambos os filtros nos diferentes eixos, bem como o *heading* é apresentado nas figuras 97 a 100.

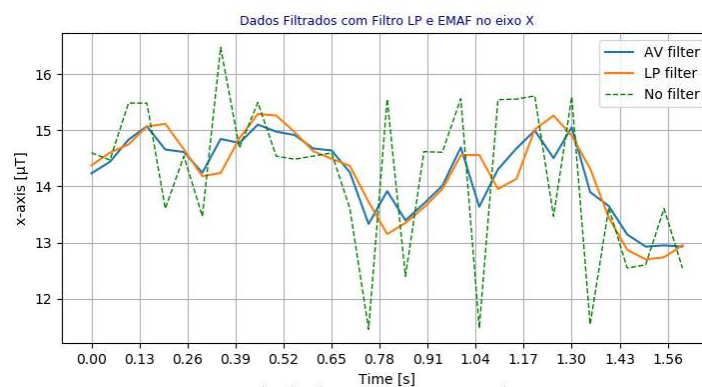


Figura 97 – Resposta temporal em regime de transição do LPF e do EMAF referente ao eixo X

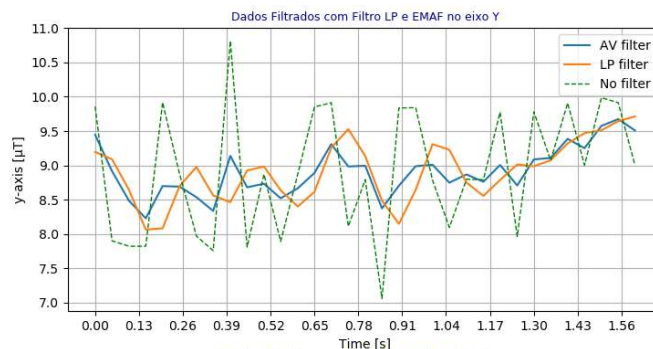


Figura 98 – Resposta temporal em regime de transição do LPF e do EMAF referente ao eixo Y

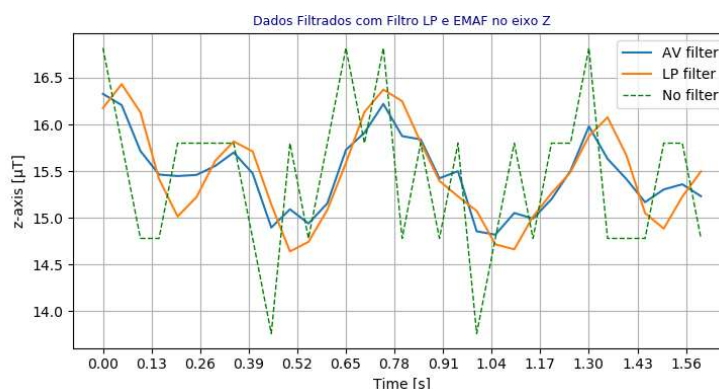


Figura 99 – Resposta temporal em regime de transição do LPF e do EMAF referente ao eixo Z

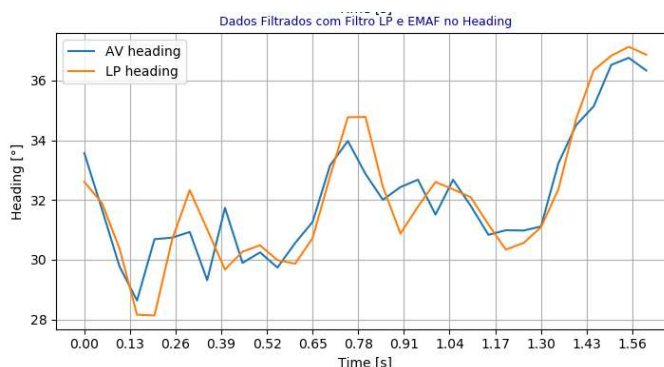


Figura 100 – Resposta temporal em regime de transição do LPF e do EMAF referente ao Heading

As figuras 101 a 104 mostram os testes realizados para o regime de transição. Realizando uma combinação entre os valores obtidos em regime estacionário e em regime de transição, nota-se que o EMAF demonstra melhor comportamento. Em regime estacionário o EMAF apresenta valores superiores do Máximo do Erro Absoluto, e da Média do Erro Absoluto, que o filtro LPF no entanto pode-se realçar que a Variância no EMA é muito menor que para o LPF. No regime de transição nota-se que o EMA apresenta um melhor desempenho na estabilização dos dados filtrados face ao atraso verificado no LPF, que por vezes originam sobre-elevação dos dados. Assim, podemos concluir que o EMAF será o filtro ótimo para atenuar o ruído presente nos dados do magnetómetro, neste cenário.

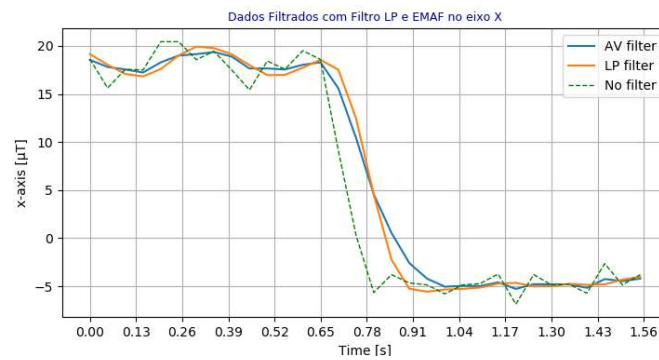


Figura 101 – Resposta temporal em regime de transição do LPF e do EMA no eixo X

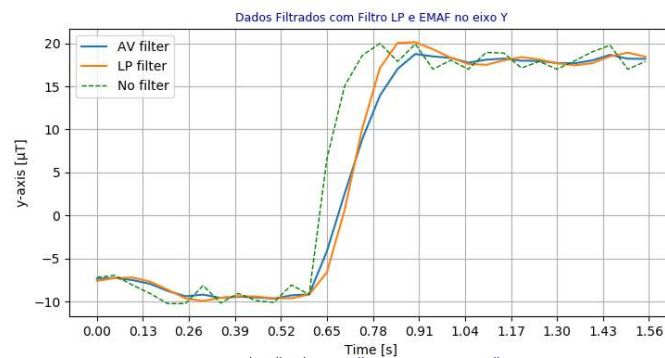


Figura 102 – Resposta temporal em regime de transição do LPF e do EMA no eixo Y

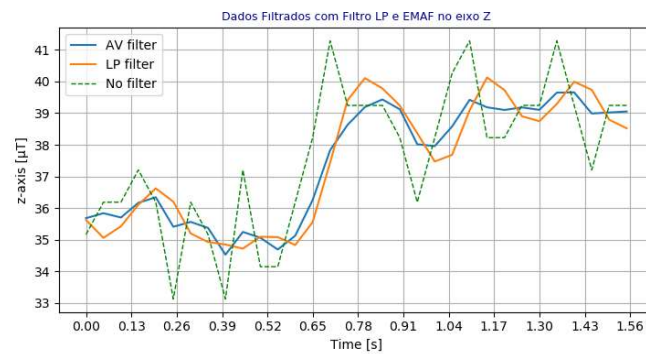


Figura 103 – Resposta temporal em regime de transição do LPF e do EMA no eixo Z

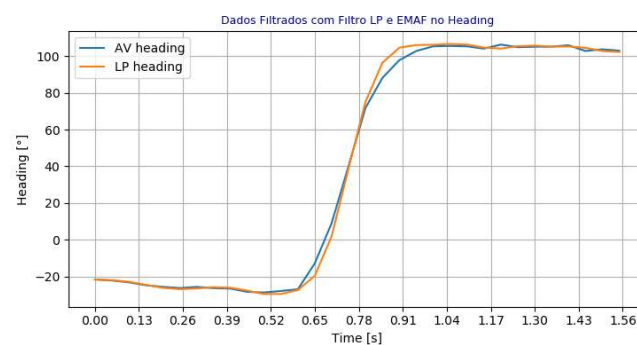


Figura 104 – Resposta temporal em regime de transição do LPF e do EMA no Heading

4.2.3 Desempenho do *Tilt*

Para verificar o desempenho do *tilt* foram realizadas duas experiências, sendo uma delas realizada num laboratório e outra no terreno.

A experiência realizada no laboratório consistiu em colocar o *Duro Inertial* sobre uma estrutura e realizar movimentos em diferentes direções, neste caso no eixo *X* e *Y*. Na Figura 105 demonstra o *Duro Inertial* na estrutura (neste caso num carrinho).

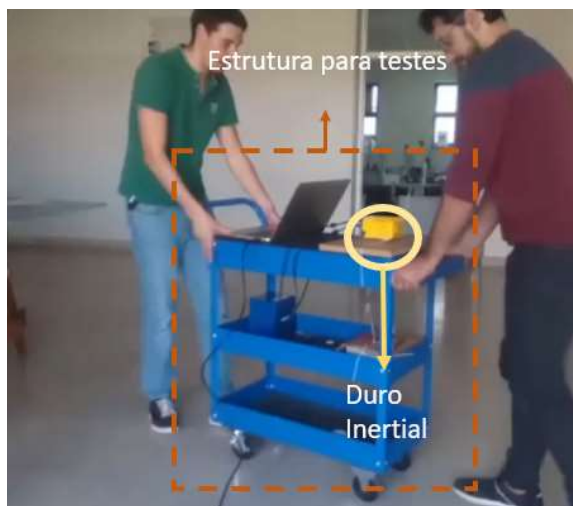


Figura 105 – Estrutura para testes do *tilt* no laboratório

Desta forma ao realizar movimentos de inclinação verificou-se a variação do *Heading*, sendo esta variação demonstrada pela seta conforme mostra a figura 106 (c). Na Figura 106 (a) verifica-se uma inclinação no eixo *X* e na figura 106 (b) é realizada uma inclinação eixo *Y*. Na Figura 106 (c) mostra a direção do *Heading*, este pelo qual se manteve estável ao longo das inclinações aferindo assim o funcionamento deste.



Figura 106 – Resultado da experiência do *tilt* no laboratório. (a): Movimento executado no eixo *X*; (b): Movimento executado no eixo *Y*; (c): Seta designada como a direção dada pelo *Heading*

Na experiência do *tilt* realizada no terreno, a máquina executou um teste prático seguindo uma trajetória linear controlada pelo sistema de teleoperação. Durante essa trajetória, a máquina subiu uma inclinação de cerca de 40 graus, conforme ilustrado na Figura 107 (b), para observar os valores registrados do *heading* utilizando o *tilt* e avaliar o seu comportamento.

A Figura 107 mostra o percurso realizado pela máquina, partindo de uma área plana, seguindo para uma zona inclinada e terminando novamente numa superfície plana. No entanto, é importante observar que devido à construção mecânica da máquina, ela pode não seguir uma trajetória exatamente linear, o que pode causar uma discrepância entre o valor de *Heading* inicial e o valor de *Heading* após a conclusão da trajetória. Além disso, o terreno acidentado onde o teste foi realizado pode resultar num leve deslizamento das lagartas do robô, afetando sua orientação. Este teste permite determinar se existem variações significativas no ângulo quando o robô está numa área inclinada. O robô iniciou o percurso com a orientação voltada para o Norte Magnético, ou seja, com um *Heading* de 90 graus.

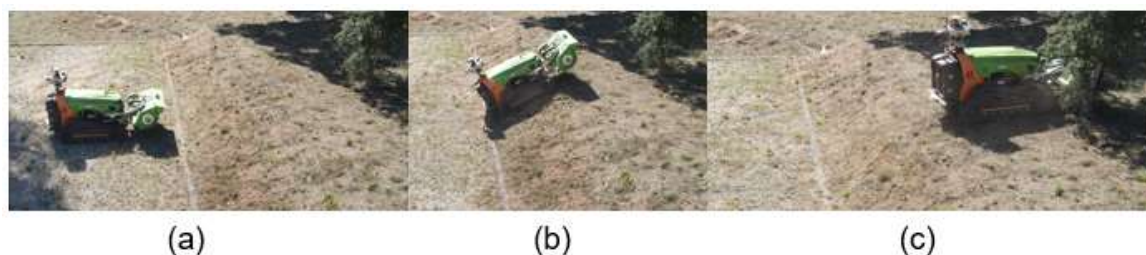


Figura 107 – Teste do Tilt. (a): Ponto inicial do robô (b) Trajetória do robô com inclinação. (c) Fim do percurso

O gráfico na Figura 108 ilustra as mudanças no *Heading* do Magnetômetro ao longo dessa trajetória. Observa-se uma variação mínima, a qual pode ser atribuída à estrutura mecânica da máquina (a máquina não se desloca propriamente em linha reta) e às vibrações geradas pelo motor de combustão interna. Essas vibrações podem, em certa medida, afetar o desempenho dos sensores. No entanto, é evidente que o *Tilt* está a funcionar e que o Filtro Complementar consegue, em certa medida, atenuar as vibrações da máquina e auxiliar na correção do cálculo do *Tilt*.

Comportamento do Heading com a compensação do Tilt

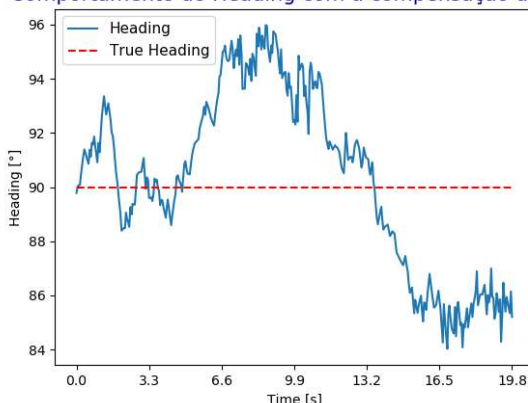


Figura 108 – Resposta temporal do Tilt durante uma trajetória em linha reta

4.3 Teste da Posição do Robô

Para avaliar a precisão da posição do robô, é fundamental determinar qual dos filtros, seja o SMAF ou o Kalman Filter, oferece o melhor desempenho. Isso foi investigado por meio de duas experiências que envolveram a execução de uma trajetória linear, durante a qual as coordenadas de GPS foram registadas. Posteriormente, essas coordenadas foram convertidas em coordenadas cartesianas, permitindo a avaliação do desvio em relação à posição de referência numa trajetória com uma distância de 10 metros. Estes testes foram designados por "Teste A" e "Teste B", sendo ambos executados à mesma distância (10 metros). Para compará-los, calculou-se a distância máxima da trajetória e a raiz quadrada da média dos quadrados *Root Mean Squared* (RMS) para todos os pontos ao longo da trajetória.

O teste A foi realizado numa estrutura fixa, um muro nivelado onde se construiu uma estrutura móvel que permitisse movimentar o *Duro Inertial* à mão ao longo desta estrutura fixa. Esta estrutura pode ser observada na Figura 109(a) onde o *Duro Inertial* está afixado à estrutura móvel feita de metal com umas rodas para que o sistema possa deslizar na estrutura fixa como se mostra na Figura 109 (c).

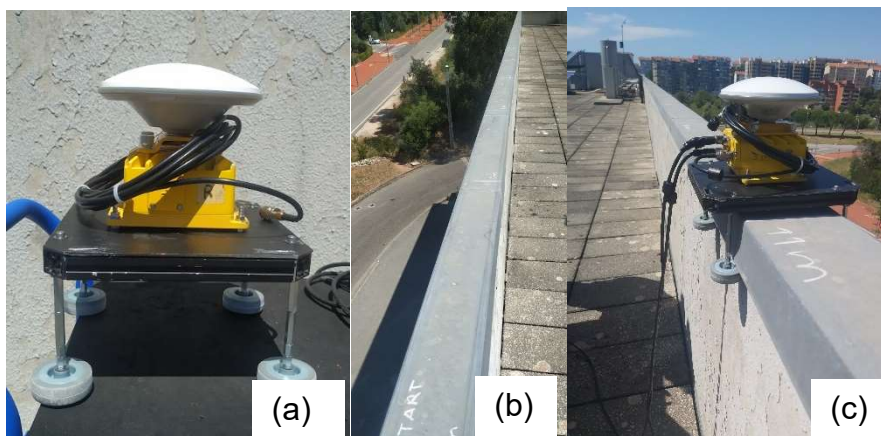


Figura 109 –Teste A. (a): estrutura móvel; (b): estrutura fixa; (c): execução do teste

Ao colocar o sistema *Duro Inertial* no muro e para verificar quanto afastado este se encontra da posição desejada foi necessário obter o *heading*, que deu -47.2° e quando representarmos os dados do GPS num gráfico este vai estar apresentado como sendo um *declive*. Os resultados do Teste A estão detalhados tanto na Tabela 5 quanto na Figura 110, que exibe uma representação gráfica da comparação dos dados.

Tabela 5 – Valores Máximo e RMS obtidos durante a trajetória do teste A

Max (m)		RMS (m)	
<i>KF</i> ^[a]	<i>SMAF</i> ^[b]	<i>KF</i>	<i>SMAF</i>
0,225	0,367	0,11	0,21

[a]: Kalman Filter

[b]: Simple Moving Average Filter

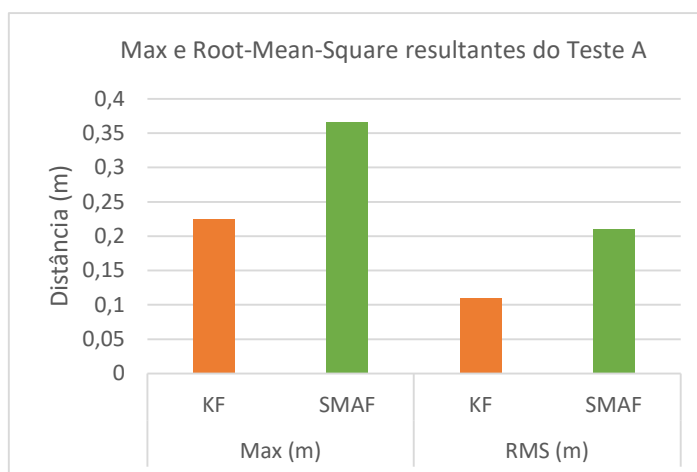


Figura 110 – Gráfico de barras comparativo dos resultados obtidos do teste A

A Figura 111 mostra graficamente os pontos adquiridos durante a trajetória, entre o início e o destino com a aplicação dos diferentes filtros para aferir a posição do sistema.

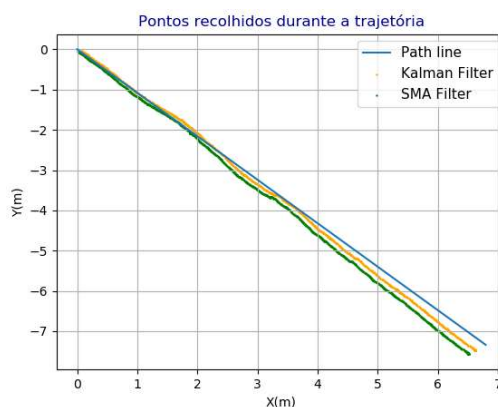


Figura 111 – Pontos coletados durante a trajetória do Teste A

Analisando os valores calculados e sua representação gráfica, podemos observar que neste teste, o Kalman Filter demonstra um desempenho superior ao Filtro SMA. Ele regista uma distância menor em relação à trajetória de referência e também exibe um menor valor de RMS, indicando que os valores estão mais próximos da trajetória desejada.

No Teste B a experiência foi realizada no terreno com a máquina florestal. Neste teste, foi selecionado um algoritmo de navegação e foi dada uma coordenada de destino ao sistema Sentry para conduzir o robô de forma autónoma até a coordenada final. Desta forma foi permitido avaliar o comportamento da posição do robô ao longo de uma trajetória específica. O robô iniciou a trajetória orientado a Este, ou seja, com um declive de 0° , e a coordenada de destino foi definida como -10.0, 0.0 para corresponder a um percurso com uma distância de 10 metros. A Figura 112 ilustra a orientação inicial do robô e os pontos de partida (Ponto A) e destino (Ponto B).



Figura 112 – Teste B realizado no terreno

Desta forma, foram repetidas três trajetórias ambas do Ponto A até ao Ponto B para comparar o desempenho dos diferentes filtros (SMA e Kalman) bem como o desempenho do sistema sem qualquer tipo de filtragem de dados.

As trajetórias foram realizadas de forma individual com cada filtro para avaliar o comportamento do robô durante a navegação autónoma com o auxílio de um algoritmo de navegação. A Figura 114 exibe visualmente as trajetórias percorridas pelo robô com cada filtro, enquanto na Figura 113 é feita uma comparação direta entre os resultados dos filtros no mesmo gráfico. Os resultados da experiência estão apresentados em detalhe na Tabela 6.

Tabela 6 – Dados obtidos da experiência no terreno

Max (m)			RMS (m)		
$KF^{[a]}$	$SMAF^{[b]}$	$NF^{[c]}$	KF	$SMAF$	NF
0,328	0,396	0,4057	0,168	0,177	0,226

[a]: Kalman Filter

[b]: Simple Moving Average Filter

[c]: Sem Filtro

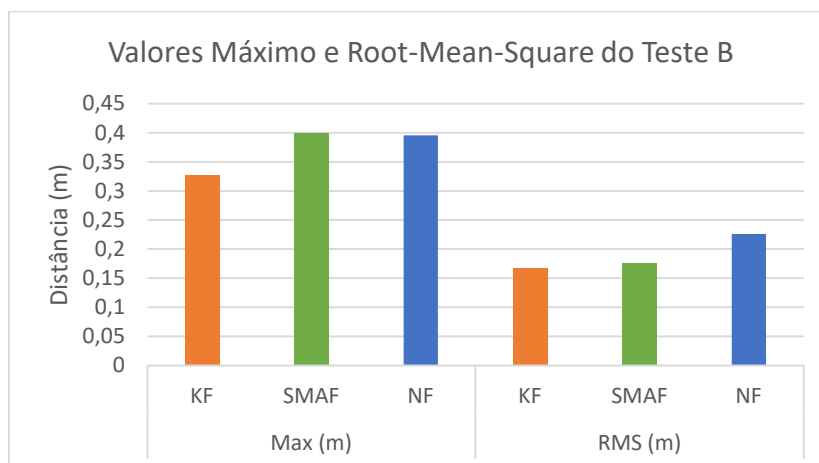


Figura 113 – Gráfico de barras comparativo dos resultados obtidos do teste B

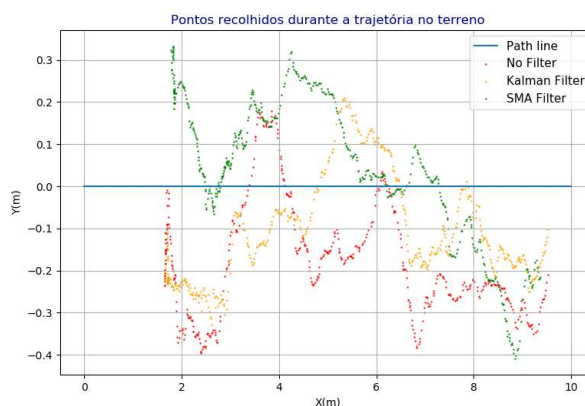


Figura 114 – Pontos coletados durante a trajetória do robô sem filtros e com o Filtro de Kalman e o Filtro SMA

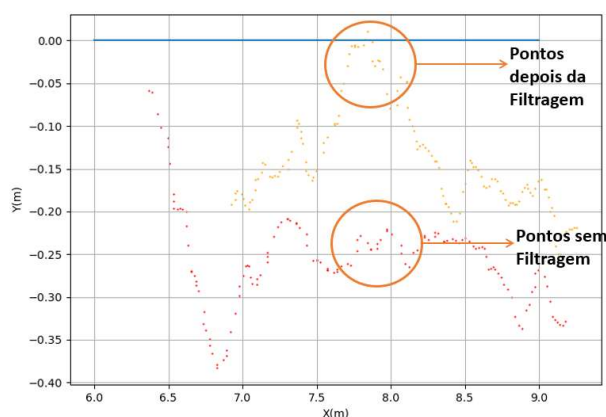


Figura 115 – Observação dos erros presentes entre o filtro de Kalman e o não uso de filtro

Os dados da Tabela 6 e do gráfico da Figura 113 oferecem uma visão clara do desempenho superior do Kalman Filter na filtragem dos dados do GPS. Notavelmente, observa-se uma redução de cerca de 7 cm no valor máximo da

distância em relação à trajetória de referência quando é utilizado filtro de Kalman em comparação com a ausência de filtragem (resultando em um cálculo de $0.4057-0.328$). Além disso, o filtro de Kalman apresenta um valor menor de RMS, o que indica uma maior confiabilidade nos resultados. O filtro SMA também exibe um valor de RMS reduzido, embora o valor máximo da distância até a trajetória seja ligeiramente maior, cerca de 6 cm.

A Figura 115 ilustra claramente como a filtragem, neste caso com o filtro de Kalman, impacta os dados em comparação com os dados não filtrados, resultando numa melhor aproximação da posição. Os resultados dos testes A e B revelam um desempenho superior do filtro de Kalman, embora no teste B os valores tenham sido ligeiramente mais elevados. Essa diferença pode ser atribuída às condições distintas nas quais ambos os filtros foram testados devido que enquanto no teste A assegurou-se o movimento retilíneo do GPS (*Duro Inertial*) no teste B não é de todo possível controlar, devido às irregularidades do terreno em que a máquina LV600 PRO não se move em linha reta, assim como a presença de vibrações da máquina no teste B que não estão presentes no teste A.

4.3 Teste ao Algoritmo Desenvolvido para o Sistema de Visão

Para analisar o desempenho do algoritmo desenvolvido para o sistema de visão a ser integrado no robô, foram realizados dois testes distintos. No primeiro teste, realizado em ambiente de laboratório, consistiu em avaliar a capacidade do algoritmo YOLOv5 a funcionar na Jetson Xavier Nx, utilizando duas câmaras térmicas e a câmara RGB da Intel, juntamente com a câmara de profundidade (*Depth*). No segundo teste, a experiência foi realizada no terreno, com a máquina operacional. Ambas as experiências tiveram como objetivo validar o desempenho dos equipamentos, especialmente a câmara RGBD Intel Realsense, enquanto observávamos como o algoritmo YOLOv5 inferia a distância até objetos, neste caso considerando pessoas como obstáculos dinâmicos.

Em relação ao primeiro teste como se pode observar na Figura 116 é apresentado o *setup* com as câmaras, bem como o funcionamento do YOLOv5 em cada uma delas. Verificou-se que a deteção de objetos a uma taxa de 100 ms permite uma boa deteção de obstáculos quando a máquina estiver em modo autónomo. Desta maneira validou-se o funcionamento do YOLOv5 onde se pode extrair a informação que interessa para ser utilizada quando a máquina estiver a executar uma trajetória de forma autónoma.

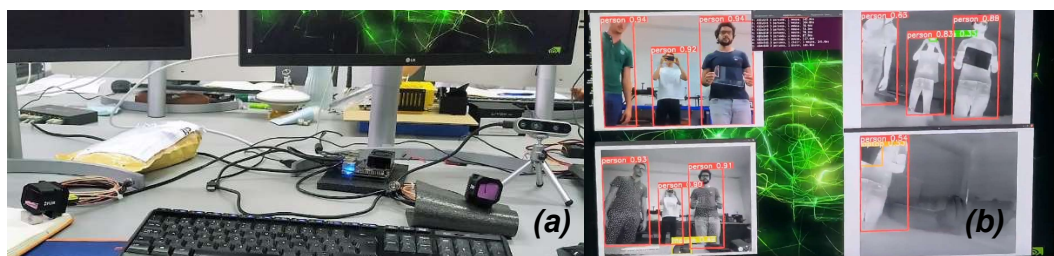


Figura 116 – Teste para verificar o funcionamento do YOLOv5. (a): *setup* com duas câmaras térmicas, uma câmara RGB e uma câmara Depth; (b): resposta do YOLOv5

O segundo teste teve como objetivo avaliar a precisão da medição de distância quando é detetada uma pessoa. Para realizar esse teste, foram projetadas marcas no terreno em distâncias estratégicas, conforme ilustrado na Figura 117. Uma pessoa ocupou cada uma dessas posições das marcas, fornecendo dados reais de distância, que foram então comparados com as leituras da câmara. As marcas foram colocadas a intervalos de 3, 6, 9, 10 e 12 metros. Foram calculadas a variância entre os dados da câmara e os valores reais obtidos a partir dessas marcações.

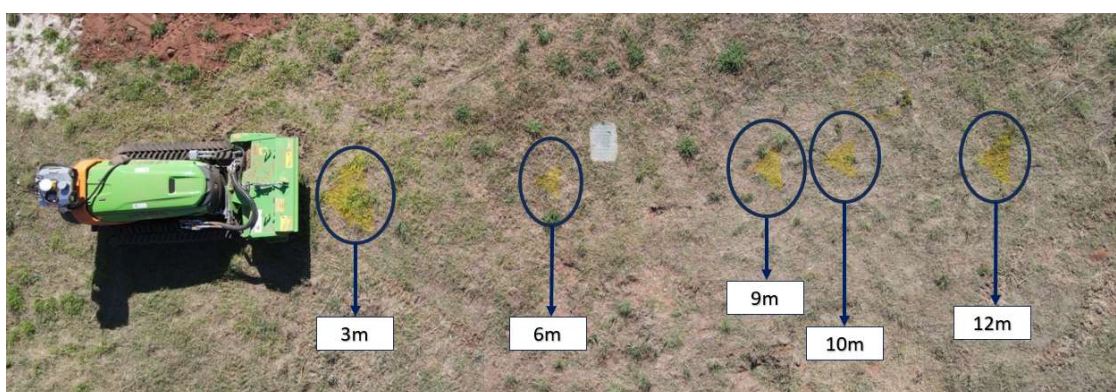


Figura 117 – Distâncias definidas para comparação dos dados reais com os dados da câmara

Para validar o *setup* no terreno, uma pessoa dirigiu-se para cada uma das marcas estratégicas onde permaneceu em cada uma delas aproximadamente 3 segundos. A Figura 118 mostra a posição de uma pessoa numa das marcas e o sistema de visão efetuou a deteção com a medição da distância.



Figura 118 – Procedimento de uma das etapas do teste do Algoritmo do sistema de visão no terreno. Neste caso a pessoa situava-se a 6 metros da câmara frontal do Sentry

A Tabela 7 apresenta uma comparação entre as medidas reais e as medidas fornecidas pela câmara. É evidente que existe uma discrepância entre as medidas reais e as medições realizadas pelo sistema de visão. Esses resultados podem ser influenciados pelas vibrações transmitidas pelo trator. No entanto, é observável que à medida que a distância aumenta, a variância também aumenta. Apesar disso, os valores obtidos parecem promissores. O Anexo G exibe as variações da distância para cada um dos marcadores.

Tabela 7 – Valores obtidos pelo teste executado no sistema de visão

<i>Distância real(m)</i>	<i>Média (m)</i>	<i>Variância (m)</i>	<i>Máximo Valor (m)</i>	<i>Mínimo Valor (m)</i>
3	2,855429	0,007638455	3,02	2,66
6	6,675714	0,141206818	7,34	5,7
9	9,778095	0,136692562	11,68	9,17
10	10,61905	0,331344628	11,68	9,5
12	12,17095	0,202512603	13,04	11,41

4.4 Teste ao Algoritmo do LIDAR

O teste realizado com o sensor LIDAR tinha como objetivo avaliar a capacidade do algoritmo *adaptive_clustering* em detetar obstáculos em um ambiente florestal. Após vários testes, determinou-se que o valor da distância " d ", mencionado na seção 3.9, deveria ser definido em 2 metros ao quadrado, a fim de evitar uma dispersão excessiva dos *clusters* na deteção de objetos. Além disso, foi estabelecido um valor de " Z_{min} " de 2 metros.

A Figura 119 ilustra a saída desse algoritmo quando deteta obstáculos, criando polígonos em torno dos objetos identificados. Em cada polígono, são registados os valores máximos e mínimos das coordenadas X e Y, juntamente com a distância até o objeto detetado. Este teste foi conduzido com a máquina em teleoperação para avaliar o impacto das vibrações no desempenho do sistema.

Constatou-se que as vibrações geradas pelo funcionamento da máquina podem causar interferências, criando objetos "fantasmas". No entanto, concluiu-se que essas falhas podem ser colmatadas por meio de um período de deteção de objetos ou pela consideração da presença ou ausência de um objeto ao longo do tempo.

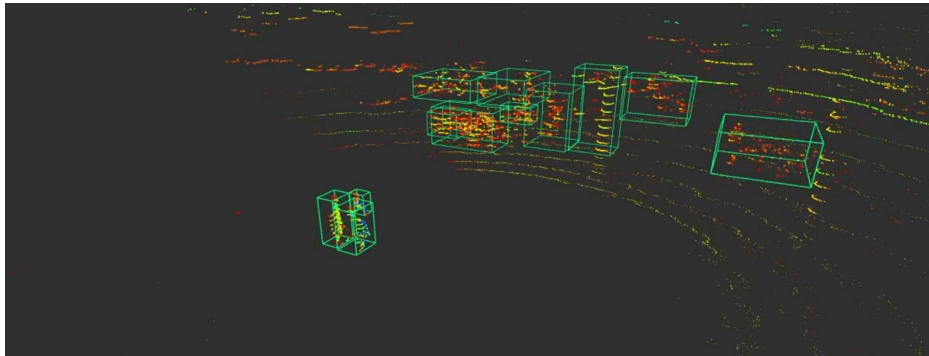


Figura 119 – Algoritmo Adaptive clustering na point cloud do Velodyne VLP-16

CAPÍTULO 5: Conclusões

O trabalho apresentado neste projeto teve como objetivo o desenvolvimento de uma solução tecnológica, que permita a navegação autónoma de uma máquina florestal, vocacionada para a limpeza de terrenos, a fim de prevenir os incêndios florestais. O principal objetivo foi dedicado ao desenvolvimento de um sistema sensorial capaz de adquirir um conjunto de parâmetros essenciais para otimizar o planeamento de trajetórias da máquina florestal em questão. O objetivo final compreendia a conceptualização e implementação de um conjunto de algoritmos que fossem de acordo com as especificações associadas à navegação autónoma da referida máquina florestal.

O estado da arte apresentado foi resultado de uma pesquisa bibliográfica mais detalhada, concentrando-se na busca por soluções que preenchessem e complementassem os requisitos que poderiam surgir neste estudo. Por outro lado, para além desta pesquisa, procedeu-se também à análise de processos que necessitam de ser aplicados para suprimir estes problemas.

Posto isto, a metodologia proposta compreendeu através de uma pesquisa relativa a sensores existentes no mercado passíveis de serem integrados numa caixa sensorial, neste caso o Sentry. De seguida, foram aplicados algoritmos, procedimentos de calibração e técnicas de fusão sensorial. Com base nessas soluções, implementou-se uma rede sensorial suscetível de ser integrada com o ROS, criando desta forma uma arquitetura que pode ser implementada em qualquer algoritmo para a navegação autónoma de qualquer robô móvel, com a finalidade descrita.

Na discussão de resultados, foram testados todos os métodos propostos no capítulo 3, a fim de averiguar a robustez do sistema sensorial, com vista à teleoperação e controlo de todos os movimentos da referida máquina florestal. Desta forma, testaram-se os métodos propostos para a aquisição de orientação do robô e da posição que foi baseada em [46], tendo-se verificado que o resultado obtido é similar ao alcançado neste estudo, aferindo assim a robustez do algoritmo de posição proposto. O sistema de visão implementado provou ser eficiente na identificação de pessoas, o que faz com que haja segurança durante o funcionamento do robô. Por outro lado, o algoritmo presente no LIDAR confirmou ser uma mais-valia para ser implementado no desvio de obstáculos durante a navegação.

Para concluir, acredita-se que, o trabalho que foi desenvolvido ao longo do presente projeto, tenha dado um contributo significativo para a aplicação de navegação autónoma em plataforma de robótica móvel genéricas, tornando-se assim a aplicação do Sentry em outras máquinas florestais trituradoras uma mais-valia.

5.1 Trabalho Futuro

Em relação ao trabalho futuro diferentes aspetos podem ser abordados, com o intuito de aumentar o desempenho do sistema sensorial, designadamente novos algoritmos e técnicas de fusão sensorial entre o LIDAR, GPS, IMU, magnetómetro e câmaras para aumentar a robustez dos dados obtidos.

Ao longo dos testes realizados verificou-se que apesar das boas características do Computador NVidia Jetson Xavier Nx, este deve de ser alterado por outro dispositivo com mais poder computacional para que futuramente se possa aplicar o Algoritmo YOLOv5 simultaneamente a todas as câmaras para efeitos de treino e identificação de outros tipos de objetos existentes na floresta, tais como: árvores, vegetação, pedras, etc.

O sistema de visão composto pelas câmaras RGBD e câmaras térmicas necessita de ser aprimorado. No que diz respeito às câmaras RGBD, neste caso a Intel Realsense D435i, apesar do seu desempenho estável, acredita-se que a aplicação da câmara Zed 2i, possa ser uma vantagem ao sistema Sentry devido a esta ter uma boa capacidade de medição até aos 20 metros o que poderá vir a ser uma vantagem significativa. Em relação às câmaras térmicas não foi desenvolvido neste projeto nenhuma técnica que permitisse fundir os dados provenientes destas para o sistema Sentry. Desta forma, como trabalho futuro pode-se proceder à integração desta no sistema, para melhorar a deteção de objetos. Por exemplo, propor a fusão entre os dados adquiridos pela FLIR com a *point cloud* do LIDAR a fim de obter não só a deteção de objetos, mas também a medição das distâncias até esse objeto.

Importa ainda realçar que este tipo de máquinas florestais possui vibrações consideráveis o que de certa forma prejudica o correto funcionamento de alguns sensores, sendo, pois, desejável adicionar um sistema de amortecimento com o intuito de mitigar a maior parte destas vibrações resultantes do funcionamento do motor de combustão e do deslocamento da mesma em terreno irregular.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] R. MORRISON, "The Environmental Impact of Wildfires," 18 Agosto 2022. [Online]. Available: <https://earth.org/environmental-impact-of-wildfires/>.
- [2] E. E. Agency, "Forest fires in Europe," 18 Nov 2021. [Online]. Available: <https://www.eea.europa.eu/ims/forest-fires-in-europe>. [Acedido em Jul 2023].
- [3] C. Ribeiro, S. Valente, C. Coelho e E. Figueiredo, "A look at forest fires in Portugal: technical," *Scandinavian Journal of Forest Research*, pp. 317-325, 2014. doi: 10.1080/02827581.2014.987160
- [4] "Fire-resistant landscaping - 6.303," Extension, <https://extension.colostate.edu/topic-areas/natural-resources/fire-resistant-landscaping-6-303/> [Acedido em Jul 2023]
- [5] S. M. Newman, R. F. Keefe, R. H. Brooks, E. Q. Ahonen e A. M. Wempe, "Human Factors Affecting Logging Injury Incidents in Idaho and the Potential for Real-Time Location-Sharing Technology to Improve Safety," *safety*, vol. 4, pp. 43-61, 2018. doi: 10.3390/safety4040043
- [6] F4F, "F4F-Forest for Future," [Online]. Available: <https://f4f.serq.pt/pt/project/pp8-demonstracao-das-tecnicas-e-ferramentas-de-limpeza-florestal>. [Acedido em 08 2023].
- [7] MDB, "LV 600 PRO," MDB, [Online]. Available: <https://www.mdb srl.com/eng/product/lv-600-pro/8/>. [Acedido em Aug 2023].
- [8] H. P. Moravec, "Britannica," The Editors of Encyclopædia Britannica, [Online]. Available: <https://www.britannica.com/technology/robot-technology>. [Acedido em Aug 2023].
- [9] R. H. o. Fame, "Robot Hall of Fame," 2003. [Online]. Available: <http://www.robothalloffame.org/inductees/03inductees/unimate.html>. [Acedido em Aug 2023].
- [10] ROBOTNIK, "History of robots and robotics," ROBOTNIK, 02 Nov 2021. [Online]. [Acedido em Jul 2023].
- [11] Abby, "Shakey the Robot Explained: Everything You Need to Know," History Computer, 25 Jul 2023. [Online]. [Acedido em Aug 2023].
- [12] U. Robots, "Universal Robots," 25 Maio 2022. [Online]. [Acedido em Jul 2023].
- [13] E. Oyekanlu, A. C. Smith, W. P. Thomas, G. Mulroy, D. Hitesh, M. Ramsey, D. J. Kuhn, J. D. Mcghinnis, S. C. Buonavita, N. A. Looper, M. Ng, A. Ng'oma, W. Liu, P. G. McBride, M. G. Shultz, C. Cerasi e D. Sun, "A Review of Recent

- Advances in Automated Guided Vehicle Technologies: Integration Challenges and Research Areas for 5G-Based Smart Manufacturing Applications,” *IEEE Access*, vol. 8, pp. 202312-202353, 2020. doi: 10.1109/ACCESS.2020.3035729
- [14] J. Billingsley, A. Visala e M. Dunn, “Robotics in Agriculture and Forestry,” em *Springer Handbook of Robotics*, 2008, pp. 1065-1077. doi: 10.1007/978-3-540-30301-5_47
- [15] ROBOTNIK, “Robotics applications in agriculture,” 31 Jan 2022. [Online]. [Acedido em Julho 2023].
- [16] Cyber-Weld, “Robots in Agriculture and Farming,” 26 Maio 2022. [Online]. [Acedido em Julho 2023].
- [17] P. Papadakis, “Terrain traversability analysis methods for unmanned ground vehicles: A survey,” *Elsivier*, vol. 26, pp. 1373-1385, 2013. doi: 10.1016/j.engappai.2013.01.006
- [18] R. Siegwart, P. Lamon, T. Estier, . M. Lauria e R. Piguët, “Innovative design for wheeled locomotion in rough terrain,” *Elsivier*, vol. 40, pp. 151-162, 2002. doi: 10.1016/S0921-8890(02)00240-3
- [19] R. G. D'Eon, R. Serrouya, . G. Smith e C. O. Kochanny, “GPS Radiotelemetry Error and Bias in Mountainous Terrain,” *JSTOR*, vol. 30, pp. 430-439, 2002.
- [20] M. S. Couceiro, D. Portugal, J. F. Ferreira e R. P. Rocha, “SEMFIRE: Towards a new generation of forestry maintenance multi-robot systems,” *2019 IEEE/SICE International Symposium on System Integration (SII)*, pp. 270-276, 29 Abril 2019. doi: 10.1109/SII.2019.8700403
- [21] A. Woo, B. Fidan, W. Melek e S. Zekavat, “Localization for Autonomous Driving,” em *Handbook of Position Location: Theory, Practice, and Advances, Second Edition, Second Edition*, Seyed A. (REZA) Zekavat, R. Michael Buehrer, 2019.
- [22] J. Diebel, “Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors,” *Matrix*, vol. 58, p. 15, 2006.
- [23] britannica, “The Britannica Dictionary,” [Online]. Available: <https://www.britannica.com/dictionary/navigation>. [Acedido em Julho 2023].
- [24] E. D. Kaplan e C. J. Hegarty, *Understanding GPS Principles and Applications*, 1996.
- [25] modelismobh, “Sistemas de posicionamento Gps, Glonass, Galileo, Beidou: conheça as diferenças,” 07 Julho 2021. [Online]. [Acedido em Julho 2023].

- [26] V. d. Topografo, “Como funcionam os Receptores GNSS?,” 16 Outubro 2021. [Online]. [Acedido em Julho 2023].
- [27] GISGeography, “Passive vs Active Sensors in Remote Sensing,” 09 Agosto 2023. [Online]. [Acedido em Agosto 2023].
- [28] T. BENNETT, “The Best Sensors for Autonomous Navigation,” 24 Setembro 2021. [Online]. [Acedido em Agosto 2023].
- [29] A. Kelly, *Modern Inertial and Satellite*, 1994.
- [30] W. Y. WONG, M. S. WONG e K. H. LO, “Clinical applications of sensors for human posture and movement analysis: A review,” *informa*, vol. 31, pp. 62-75, 2007. doi: 10.1080/03093640600983949
- [31] wikipedia, “Sensor de efeito Hall,” Maio 2014. [Online]. Available: https://pt.wikipedia.org/wiki/Sensor_de_efeito_Hall. [Acedido em Julho 2023].
- [32] wikipedia, “Magnetometro fluxgate,” 12 Fevereiro 2022. [Online]. [Acedido em Julho 2023].
- [33] M. I. Ribeiro, “Sensores em Robótica,” em *Enciclopedia Activa Multimedia*, 2004, pp. 228-229.
- [34] MaxBotix, “How Ultrasonic Sensors Work,” 01 Março 2023. [Online]. [Acedido em Julho 2023].
- [35] M. Moses, “2D LIDAR Versus 3D LIDAR,” *lidarandradar*, 2022 Junho 2021. [Online]. Available: <https://lidarandradar.com/2d-lidar-versus-3d-lidar/>. [Acedido em Julho 2023].
- [36] V. Lidar, “What is lidar?,” [Online]. Available: <https://velodynelidar.com/what-is-lidar/>.
- [37] L. Osborn, “This or That: RGB vs. Multispectral Sensors,” 2023 Janeiro 18. [Online]. [Acedido em Julho 2023].
- [38] framos, “WHAT ARE DEPTH-SENSING CAMERAS AND HOW DO THEY WORK?,” 2023 Abril 14. [Online]. [Acedido em Julho 2023].
- [39] T. FLIR, “How Do Thermal Cameras Work?,” 2023 Junho 16. [Online]. [Acedido em Agosto 2023].
- [40] S. W. Smith, *Digital Signal Processing*, San Diego, California, 1999.
- [41] Y. Verma, “A Guide To Audio Data Preparation Using TensorFlow,” 2021 Julho 6. [Online]. Available: <https://analyticsindiamag.com/a-guide-to-audio-data-preparation-using-tensorflow/>. [Acedido em Julho 2023].

- [42] ni, "IIR Filters and FIR Filters," 06 Abril 2023. [Online]. Available: https://www.ni.com/docs/en-US/bundle/diadem/page/genmaths/genmaths/calc_filterfir_iir.htm. [Acedido em Julho 2023].
- [43] L. Tan e J. Jiang, *Digital Signal Processing Fundamentals and Applications*, 225 Wyman Street, Waltham, MA 02451, USA, 2013.
- [44] A. C. Salian, "Moving Average Filter: Towards Signal Noise Reduction," 31 Maio 2022. [Online]. [Acedido em Maio 2023].
- [45] R. Hostettler e S. Sarkka, *Basics of Sensor Fusion*, 2020.
- [46] S. Han, Q. Zhang e H. Noh, "Kalman Filtering of DGPS Positions for a Parallel Tracking Application," *Transactions of the ASAE. American Society of Agricultural Engineers*, vol. 45, pp. 553-559, 2002. doi: 10.13031/2013.8856
- [47] Q. Li, L. Ranyang, K. Ji e W. Dai, "Kalman Filter and Its Application," em *2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS)*, 2015. doi: 10.1109/ICINIS.2015.35
- [48] R. G. Valenti, I. Dryanovski e J. Xiao, "Keeping a Good Attitude: A Quaternion-Based Orientation Filter for IMUs and MARGs," *sensors*, vol. 15, pp. 19302-19330, 2015. doi: 10.3390/s150819302
- [49] Y. Li e Y. Zhang, "Application Research of Computer Vision Technology in Automation," *2020 International Conference on Computer Information and Big Data Applications (CIBDA)*, Abril 2020. doi: 10.1109/CIBDA50819.2020.00090
- [50] H. Bandyopadhyay, "What Is Computer Vision? [Basic Tasks & Techniques]," 09 Junho 2022. [Online]. [Acedido em Junho 2023].
- [51] J. Terven e D. Cordova-Esparza, "A Comprehensive Review of YOLO: From YOLOv1 and Beyond," 2023. doi: 10.48550/arXiv.2304.00501
- [52] [1] R. Xu, H. Lin, K. Lu, L. Cao, and Y. Liu, "A forest fire detection system based on Ensemble Learning," *Forests*, vol. 12, no. 2, p. 217, 2021. doi:10.3390/f12020217
- [53] M. Quigley, B. Gerkey e W. D. Smart, *Programming Robots with ROS*, United States of America., 2015.
- [54] A. Anwar, "Part 2: 7 Simple Steps to Create and Build Your First ROS Package," 09 Fevereiro 2021. [Online]. Available: <https://medium.com/swlh/7-simple-steps-to-create-and-build-our-first-ros-package-7e3080d36faa>. [Acedido em Julho 2023].
- [55] automaticaddison, "Launch the Turtlesim Robot Simulation in ROS," 26

- Outubro 2019. [Online]. Available: <https://automaticaddison.com/launch-turtlesim-robot-simulation-in-ros/>. [Acedido em Julho 2023].
- [56] IGVC, 29 Novembro 2017. [Online]. Available: <https://ras.ece.utexas.edu/2017/11/29/igvc.html>. [Acedido em Julho 2023].
- [57] NVidia, “NVIDIA Jetson Xavier,” [Online]. Available: <https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-xavier-series/>. [Acedido em Junho 2023].
- [58] Intel, “Intel® NUC 11 Pro Mini PC NUC11TNKv7,” [Online]. Available: <https://ark.intel.com/content/www/us/en/ark/products/205608/intel-nuc-11-pro-mini-pc-nuc11tnkv7.html>. [Acedido em Junho 2023].
- [59] TELTONIKA, “RUT360 LTE CAT6 INDUSTRIAL CELLULAR ROUTER,” [Online]. Available: <https://teltonika-networks.com/products/routers/rut360>. [Acedido em Junho 2023].
- [60] R. Moeller, T. Deemyad e A. Sebastian, “Autonomous Navigation of an Agricultural Robot Using RTK GPS and Pixhawk,” em *2020 Intermountain Engineering, Technology and Computing (IETC)*, Utah Valley University, UT, USA, 2020. doi: 10.1109/IETC47856.2020.9249176
- [61] swiftnav, “Duro® Inertial PRODUCT SUMMARY,” 08 Agosto 2022. [Online]. Available: https://www.swiftnav.com/sites/default/files/duro_inertial_product_summary.pdf. [Acedido em Maio 2023].
- [62] V. Tadic, A. Toth, Z. Vizvari, M. Klincsik, Z. Sari, P. Sarcevic, J. Sarosi e I. Biro, “Perspectives of RealSense and ZED Depth Sensors for Robotic Vision Applications,” *machines*, vol. 10, pp. 183-209, 2022. doi: 10.3390/machines10030183
- [63] A. S. Mohammed, A. Amamou, F. K. Ayevide, S. Kelouwani, K. Agbossou e N. Zioui, “The Perception System of Intelligent Ground Vehicles in All Weather Conditions: A Systematic Literature Review,” *sensors*, vol. 20, pp. 6532-6566, 2020. doi: 10.3390/s20226532
- [64] T. FLIR, “Thermal Stereo Solutions for Seeing Through Dust in Rugged Environments,” 22 Junho 2021. [Online]. Available: <https://www.flir.eu/discover/cores-components/thermal-stereo-solutions-for-seeing-through-dust-in-rugged-environments/>. [Acedido em Maio 2023].
- [65] V. Lidar, “Velodyne - VLP-16 User Manual,” 2019. [Online]. Available: <https://velodynelidar.com/wp-content/uploads/2019/12/63-9243-Rev-E-VLP-16-User-Manual.pdf>. [Acedido em Maio 2023].

- [66] szenergy, “duro_gps_driver,” [Online]. Available: https://github.com/szenergy/duro_gps_driver. [Acedido em Maio 2023].
- [67] Ros-drivers, “velodyne,” [Online]. Available: <https://github.com/ros-drivers/velodyne>. [Acedido em Maio 2023].
- [68] ros-drivers, “roserial,” [Online]. Available: <https://github.com/ros-drivers/roserial>. [Acedido em Maio 2023].
- [69] J.-H. Wang e Y. Gao, “A new magnetic compass calibration algorithm using neural networks,” *Measurement Science and Technology*, vol. 17, pp. 153-160, 2006. doi: 10.1088/0957-0233/17/1/025
- [70] A. Kuncar, M. Sysel e T. Urbanek, “Calibration of Triaxial Accelerometer and Triaxial Magnetometer for Tilt Compensated Electronic Compass,” em *Automation Control Theory Perspectives in Intelligent Systems*, 2016, pp. 45-52. doi: 10.1007/978-3-319-33389-2_5
- [71] I. Mateos, J. Ramos-Castro e A. Lobo, “Low-frequency noise characterization of a magnetic field monitoring system using an anisotropic magnetoresistance,” *Sensors and Actuators A Physical*, vol. 235, pp. 57-63, 2015. doi: 10.1016/j.sna.2015.09.021
- [72] E. Grahn, Evaluation of MEMS accelerometer and gyroscope for orientation tracking nutrunner functionality, 2017.
- [73] CCNYRoboticsLab, “imu_tools,” [Online]. Available: https://github.com/CCNYRoboticsLab/imu_tools. [Acedido em Maio 2023].
- [74] swri-robotics, “gps_umd,” [Online]. Available: https://github.com/swri-robotics/gps_umd.
- [75] swiftnav, “Duro Inertial User Manual,” 13 Julho 2022. [Online]. Available: [https://www.swiftnav.com/resource-files/Duro%20Inertial/v2/Manual/Duro%20Inertial%20User%20Manual%20\[UM-110008-02\].pdf](https://www.swiftnav.com/resource-files/Duro%20Inertial/v2/Manual/Duro%20Inertial%20User%20Manual%20[UM-110008-02].pdf). [Acedido em Maio 2023].
- [76] ros-planning, “robot_pose_ekf,” [Online]. Available: https://github.com/ros-planning/robot_pose_ekf. [Acedido em Maio 2023].
- [77] ultralytics, “yolov5,” [Online]. Available: <https://github.com/ultralytics/yolov5>. [Acedido em Maio 2023].
- [78] pytorch, “PyTorch,” [Online]. Available: <https://pytorch.org/>.
- [79] NVidia, “NVIDIA cuDNN,” [Online]. Available: <https://developer.nvidia.com/cudnn>.

- [80] opencv, “opencv,” [Online]. Available: <https://opencv.org/>.
- [81] Numpy, “Numpy,” [Online]. Available: <https://numpy.org/>.
- [82] IntelRealSense, “librealsense,” [Online]. Available: <https://github.com/IntelRealSense/librealsense/tree/master>.
- [83] Z. Yan, T. Duckett e . N. Nicola Bellotto , “Online learning for 3D LiDAR-based human detection: experimental analysis of point cloud clustering and classification methods,” *Autonomous Robots*, vol. 44, pp. 147-164, 2020. doi: 10.1007/s10514-019-09883-y
- [84] yzrobot, “adaptive_clustering,” [Online]. Available: https://github.com/yzrobot/adaptive_clustering.

ANEXOS

Anexo A

Propriedades	Nvidia Jetson Xavier NX	Intel NUC 1185G7
GPU	384 NVIDEA CUDA Cores e 48 Tensor Cores	Intel Iris Xe Graphics G7 96Eus
CPU	6-Core NVIDIA Carmel ARMv8.2 64bit	4.8 GHz (Quad-core Intel Core i7)
Memória RAM	16GB LPDDR4x	16 GB DDR4
SSD	256GB	512 GB
Consumo de Energia	15W	28W
Dimensões	100 mm x 87 mm x 16 mm	117 mm x 112 mm x 51 mm

Anexo B

Propriedades	RUT360	DWR-921
Tecnologia de Rede	4G LTE	4G LTE
Velocidade Download	up to 300 Mbps	up to 150 Mbps
Velocidade Upload	up to 50 Mbps	up to 50 Mbps
Número de portas LAN	2	4
Número de portas WLAN	1	1
Wi-Fi	Sim, 802.11 b/g/n	Sim, 802.11n
SIM Card Slots	1	1
Recursos de Segurança	Firewall, VPN, WPA2-PSK	Firewall, VPN, WPA2-PSK
Dimensões	115mm x 95mm x 44mm	190mm x 116 mm x 22.4 mm
Peso	200g	270g
Consumo de Energia	10.5 W	12W

Anexo C

Propriedades	Intel RealSense D435i	ZED 2i
Resolução	1280 x 720 (RGB)	2x 1920 x 1080 (RGB)
Alcance Profundidade	0.2m - 10m	0.2m - 20m
Campo de Visão	91° × 65° × 100°	90° x 59° x 110°
Tracking	6-DoF Pose Tracking	6-DoF Pose Tracking
IMU	Integrated IMU	Integrated IMU
Conectividade	USB 3.0	USB 3.0
Suporte	Intel RealSense SDK	ZED SDK
Compatibilidade	Windows, Linux, macOS	Windows, Linux, macOS

Anexo D

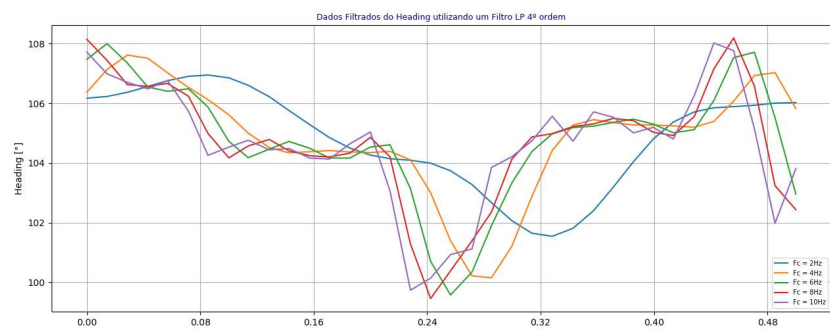
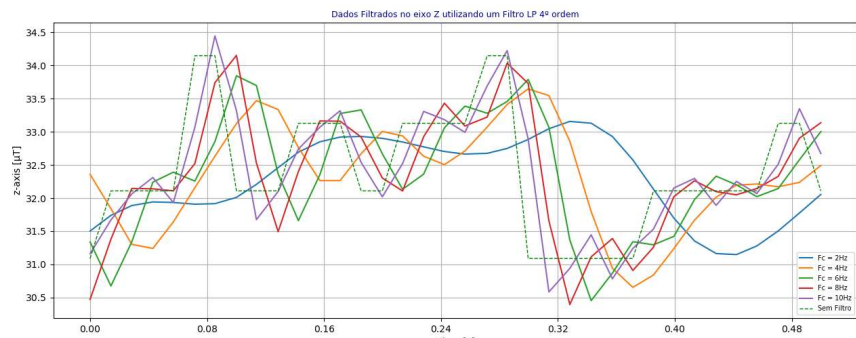
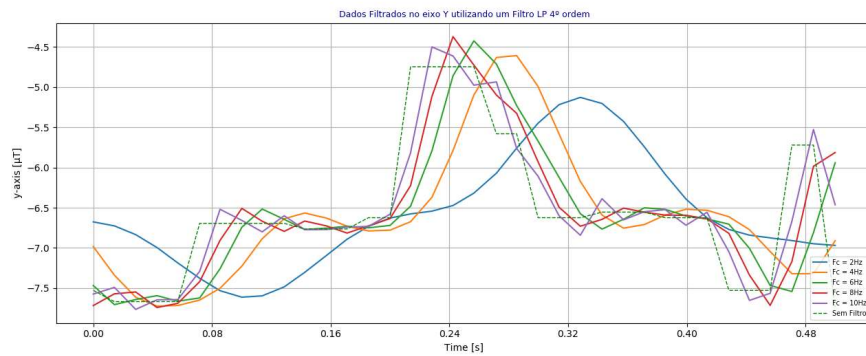
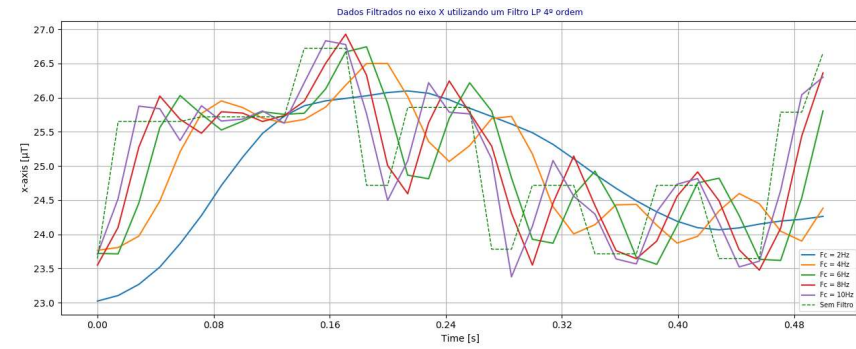
Máximo do erro Absoluto						
	Ordem	Wn=2Hz (uT)	Wn=4Hz (uT)	Wn=6Hz (uT)	Wn=8Hz (uT)	Wn=10Hz (uT)
Eixo X	2	1.82518948	2.56982448	2.93904692	2.8844926	3.10187194
	4	1.83484706	1.54687897	1.74667545	1.88382464	1.75445955
	6	1.32979403	1.88734721	2.22358845	2.03084648	2.08959478
	8	1.20984059	3.24420453	3.86667665	4.48852725	4.43780062
Eixo Y	2	0.67504644	1.16524307	1.32364062	1.30492041	1.51670833
	4	1.45654751	2.01483217	2.1767171	2.19928452	2.05833449
	6	1.32979403	1.88734721	2.22358845	2.03084648	2.08959478
	8	1.20984059	3.24420453	3.86667665	4.48852725	4.43780062
Eixo Z	2	0.67504644	0.97323757	1.26200459	1.45865221	1.69666069
	4	1.13105438	1.67471545	1.87449569	1.96126311	2.05720633
	6	0.72697818	1.8958521	2.51033532	2.55620489	2.70173584
	8	1.13028365	1.34928019	2.15306729	2.11313434	2.31904556
Média de todos os eixos	2	1.13758453	1.56943504	1.84156404	1.8826884	2.10508032
	4	1.47414965	1.74547553	1.93262941	2.01479076	1.95666679
	6	1.0736968	1.61540806	1.99626702	2.01419733	2.15686246
	8	0.93044299	1.7842296	2.28090461	2.54698492	2.59782465

Média do erro Absoluto						
	Ordem	Wn=2Hz (uT)	Wn=4Hz (uT)	Wn=6Hz (uT)	Wn=8Hz (uT)	Wn=10Hz (uT)
Eixo X	2	0.6302382	0.8609448	0.91919809	0.9194373	0.89364165
	4	0.82833596	0.77826187	0.8264261	0.84244302	0.8412791
	6	0.59876365	0.77594181	0.93343987	0.99462707	1.03005267
	8	0.75340464	1.37004872	1.40695069	1.46868336	1.47205105
Eixo Y	2	0.27497595	0.35446524	0.39519451	0.41526708	0.43085736
	4	0.54370317	0.55831061	0.58105866	0.5979368	0.58693981
	6	0.393591	0.44433715	0.54605162	0.55346273	0.57647351
	8	0.23281882	0.34788098	0.46002109	0.46342233	0.48610358
Eixo Z	2	0.43982316	0.54477238	0.59976102	0.60709253	0.63714847
	4	0.54772557	0.62456411	0.71361951	0.74154873	0.73587053
	6	0.27729611	0.6896869	0.8387466	0.84233936	0.81626943
	8	0.67042624	0.66681277	0.76717065	0.85372937	0.84879286
Média de todos os eixos	2	0.44834577	0.58672747	0.63805121	0.64726564	0.65388249
	4	0.63992157	0.6537122	0.70703475	0.72730952	0.72136315
	6	0.42321692	0.63665529	0.77274603	0.79680972	0.80759854
	8	0.55221657	0.79491416	0.87804748	0.92861169	0.93564917

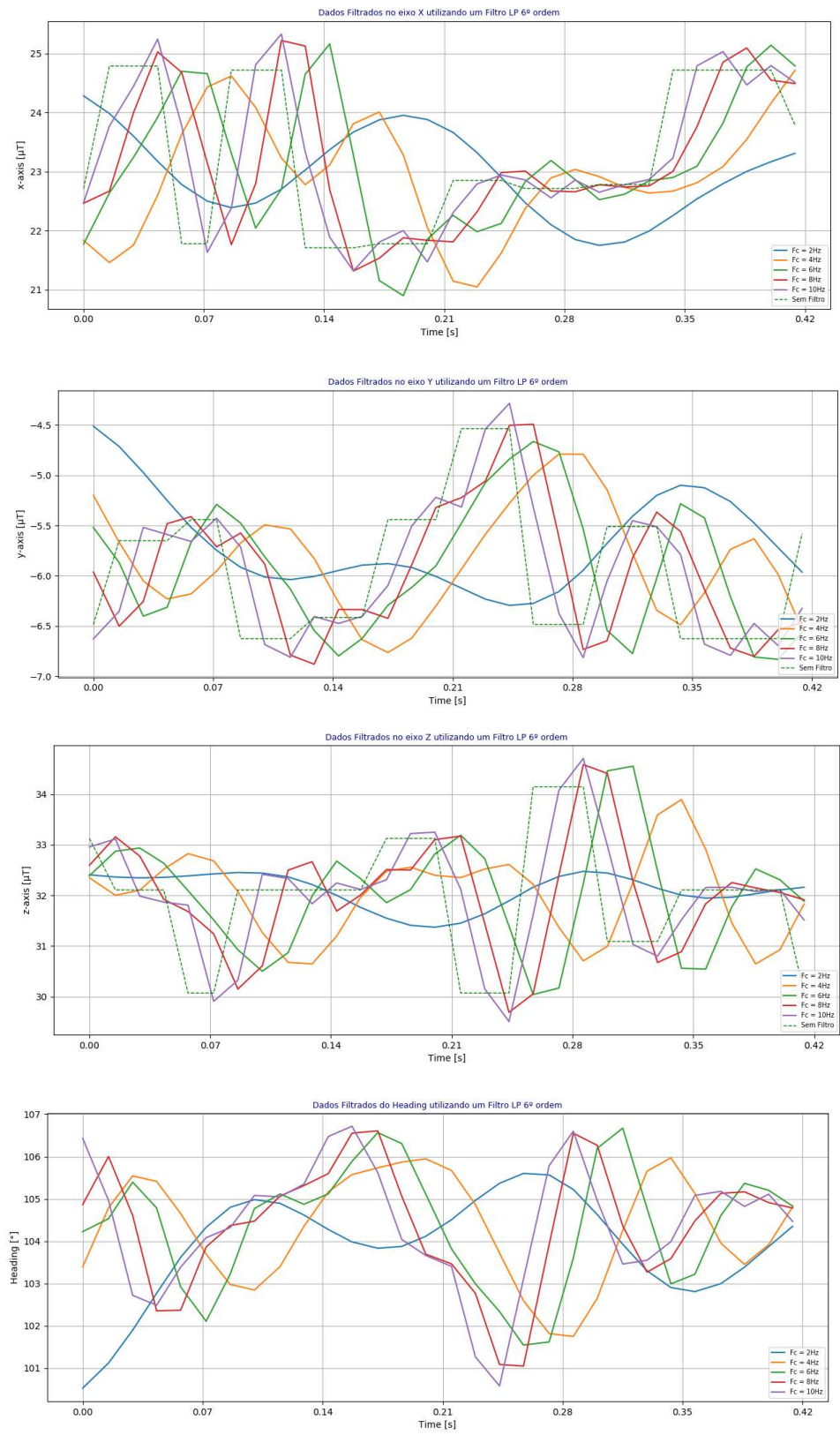
Variância						
	Ordem	Wn=2Hz (uT)	Wn=4Hz (uT)	Wn=6Hz (uT)	Wn=8Hz (uT)	Wn=10Hz (uT)
Eixo X	2	0.74803872	1.23405107	1.45529055	1.53383672	1.5326698
	4	1.04555018	1.00546695	1.23323503	1.35117799	1.4302585
	6	0.51331209	0.99257847	1.37265428	1.42275417	1.47357732
	8	0.69775501	2.67809353	2.79069969	3.11362211	3.32610673
Eixo Y	2	0.11550713	0.20558392	0.28058332	0.34874372	0.38827814
	4	0.48811209	0.64996193	0.68936634	0.70936196	0.7068456
	6	0.22462187	0.29728515	0.41841191	0.44207884	0.45678313
	8	0.07158961	0.17524716	0.26803746	0.27251665	0.28880939
Eixo Z	2	0.27032075	0.46246618	0.54991939	0.59686548	0.61790776
	4	0.63733876	0.75836418	0.91111472	0.9827569	0.99414091
	6	0.11584216	0.73422022	1.2197235	1.2944289	1.32099182
	8	0.54447632	0.5897971	0.8972039	1.02337538	1.02742597
Média de todos os eixos	2	0.37795554	0.63403372	0.76193109	0.82648197	0.84628523
	4	0.63992157	0.6537122	0.70703475	0.72730952	0.72136315
	6	0.28459204	0.67469461	1.00359656	1.0530873	1.08378409
	8	0.43794032	1.1477126	1.31864702	1.46983805	1.54744736

Anexo E

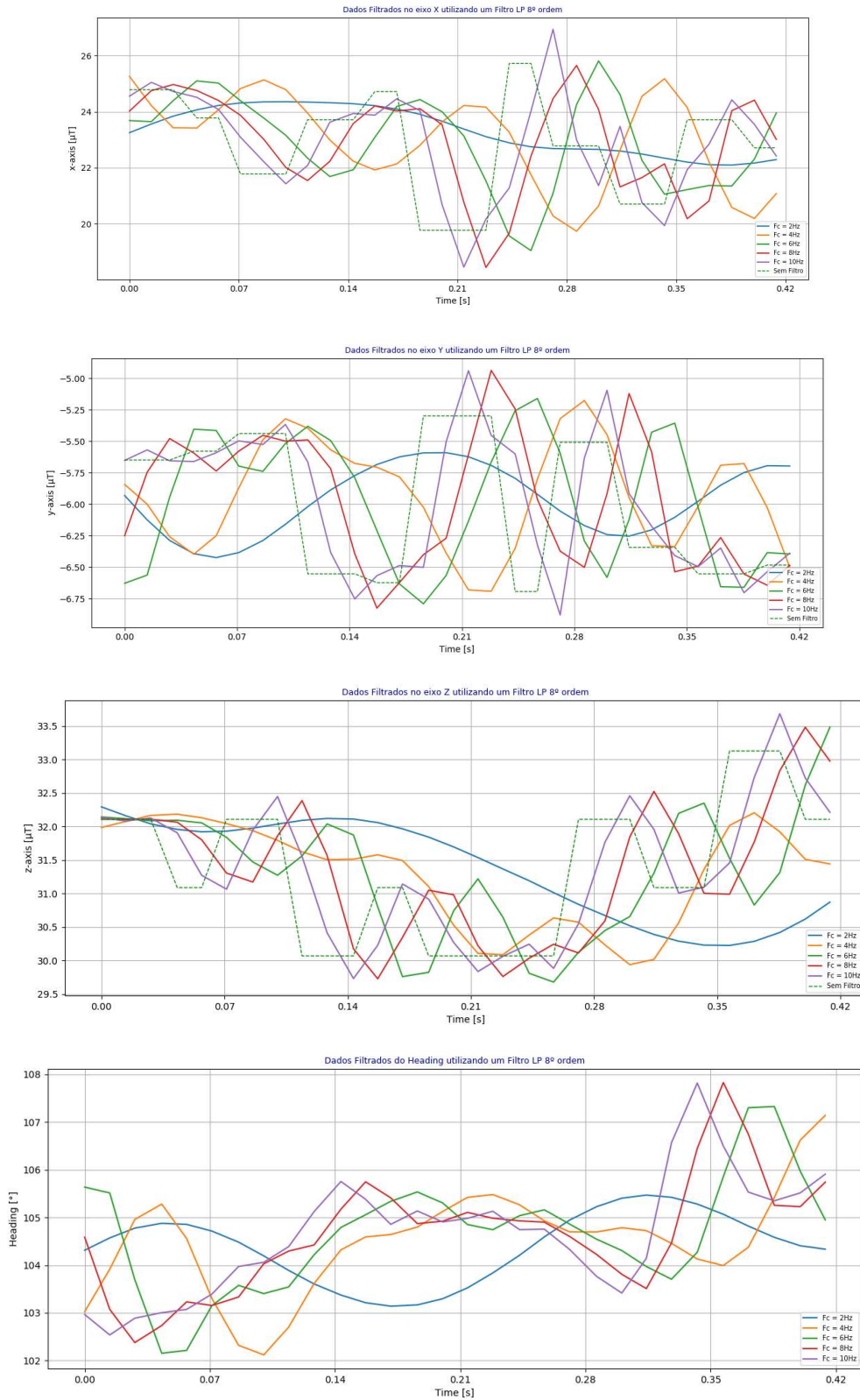
Filtro Passa-Baixo de 4º ordem:



Filtro Passa-Baixo 6º ordem:

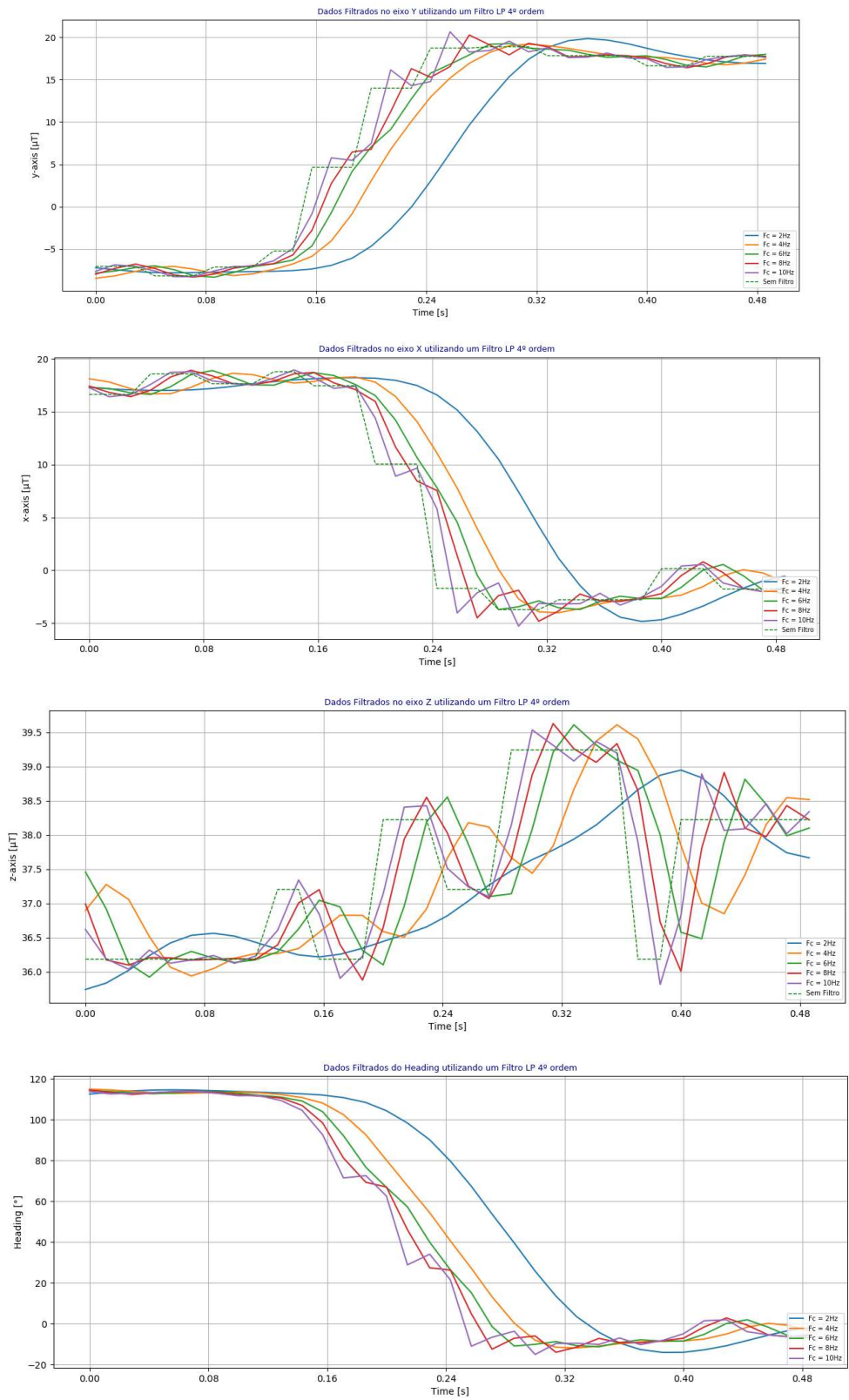


Filtro Passa-baixo 8º ordem:

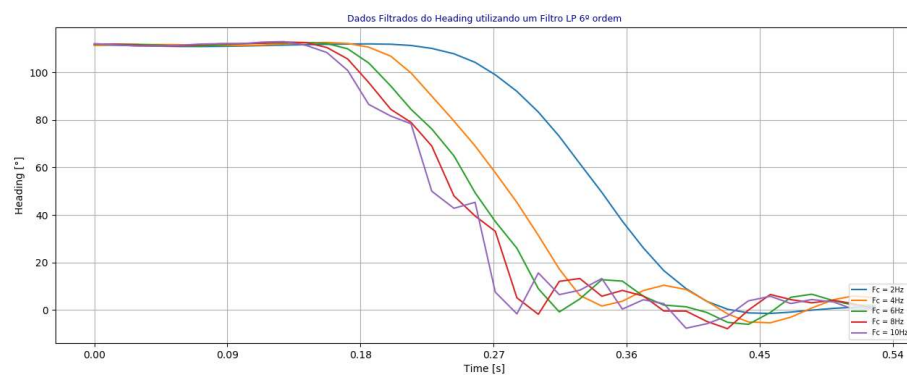
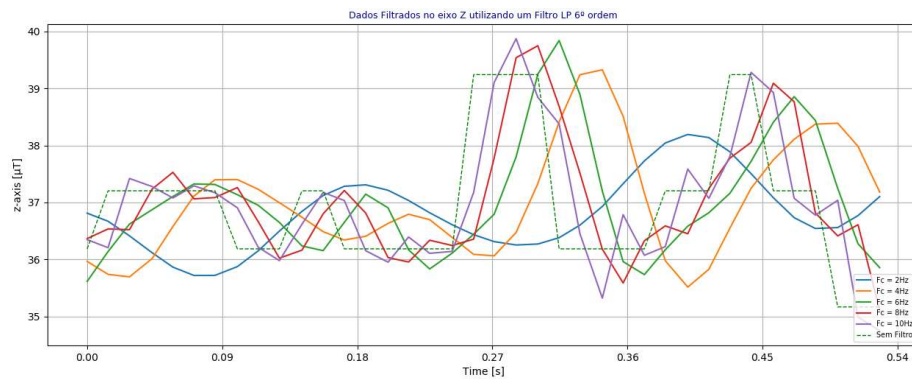
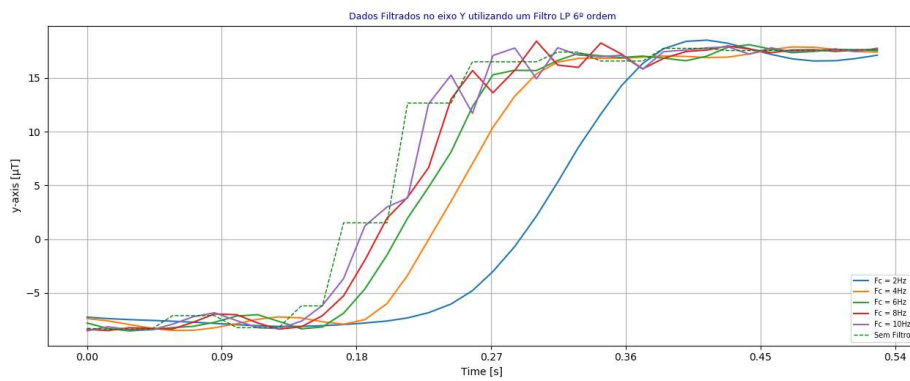
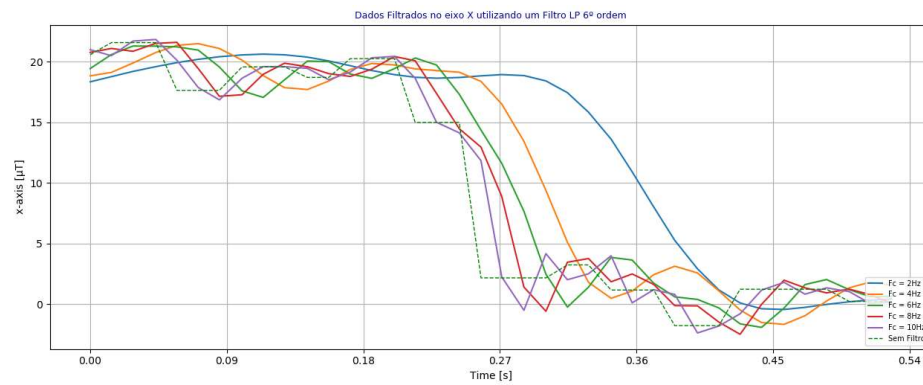


Anexo E

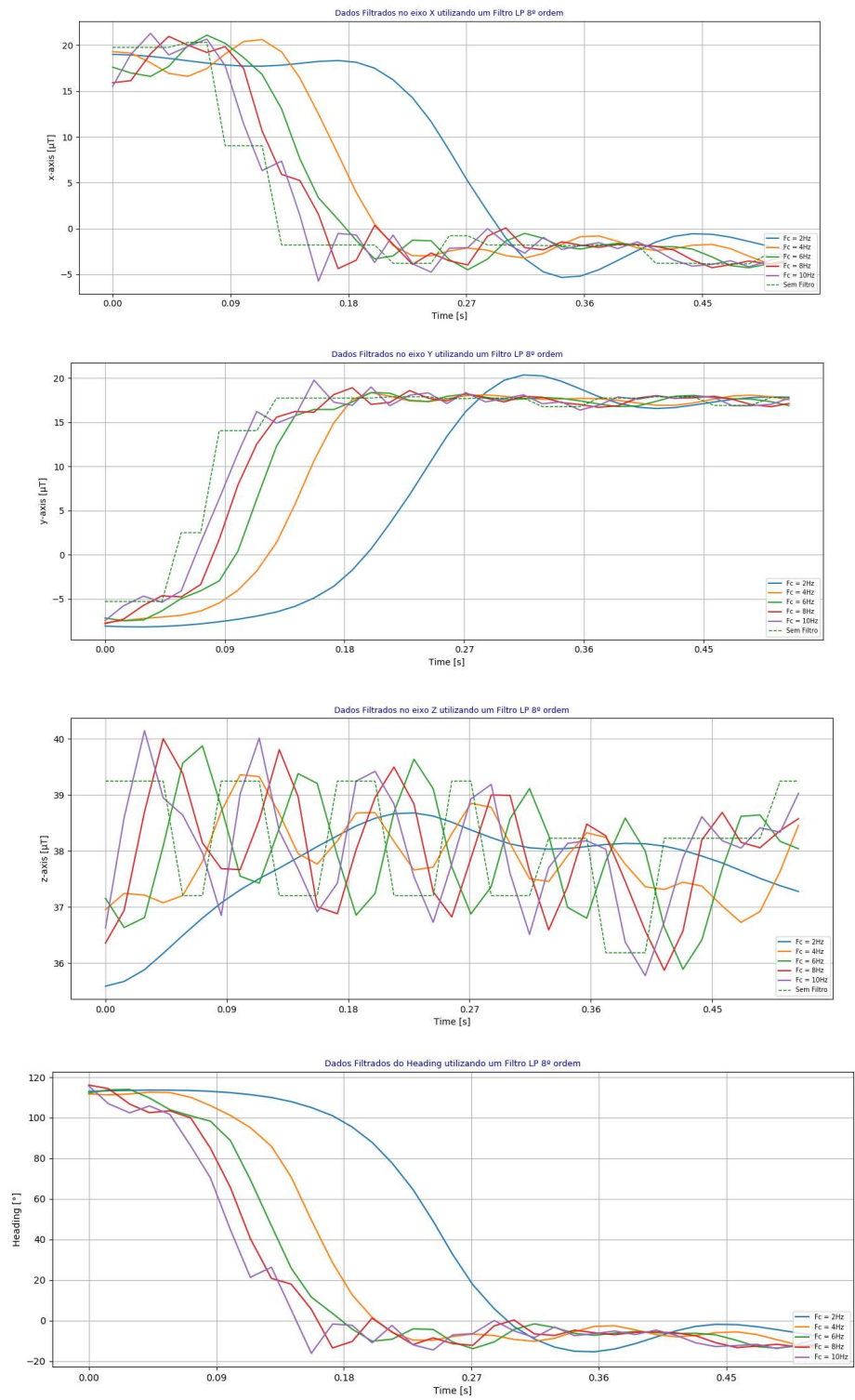
Filtro Passa-Baixo de 4º ordem:



Filtro Passa-Baixo de 6º ordem:



Filtro Passa-Baixo de 8º ordem:



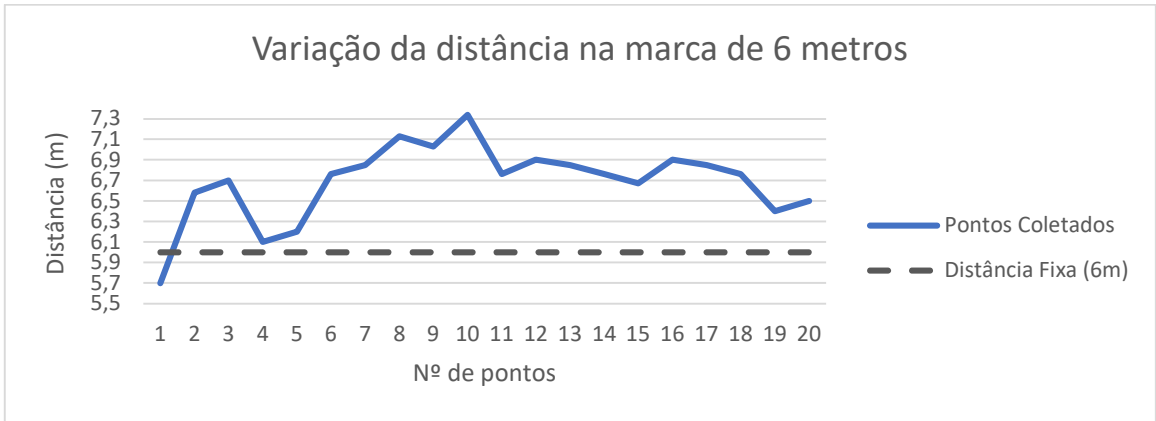
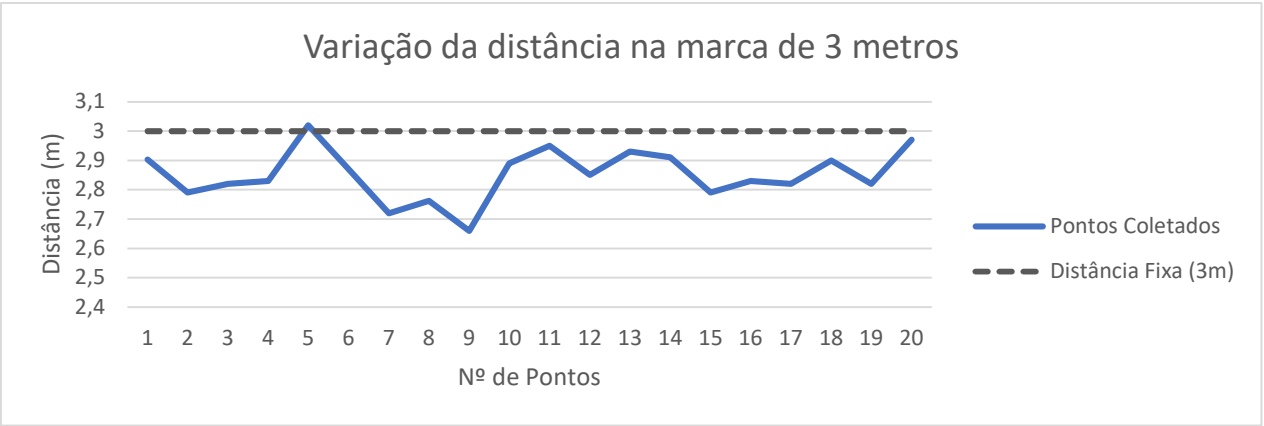
Anexo F

Máximo do erro Absoluto					
	Buffer=5 (uT)	Buffer=10 (uT)	Buffer=15 (uT)	Buffer=20 (uT)	Buffer=25 (uT)
Eixo X	2.06819667	1.40239477	1.3793548	1.11401581	0.98772613
Eixo Y	0.79891431	0.64381533	0.42890128	0.32511257	0.31283571
Eixo Z	1.84788501	0.87692575	0.71869209	0.67337252	0.55199425
Média de todos os eixos	1.57166533	0.974378624	0.84231606	0.70416697	0.6175187

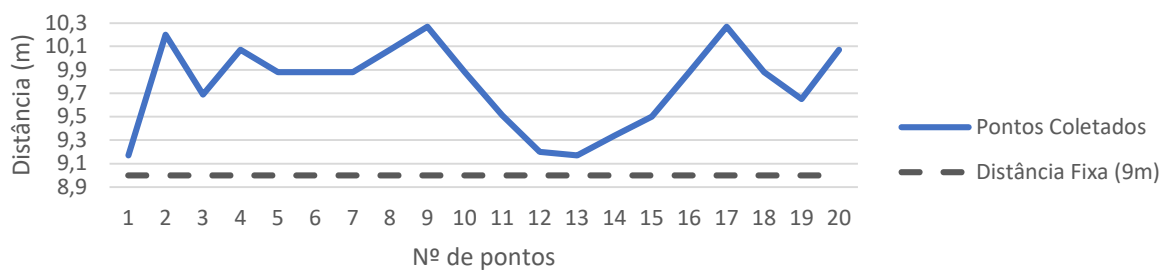
Média do erro Absoluto					
	Buffer=5 (uT)	Buffer=10 (uT)	Buffer=15 (uT)	Buffer=20 (uT)	Buffer=25 (uT)
Eixo X	0.9739544	0.69916187	0.6091091	0.52323273	0.43328555
Eixo Y	0.36100257	0.25582964	0.18797542	0.1445673	0.13592002
Eixo Z	0.60714456	0.34419056	0.34972214	0.312415	0.24931816
Média de todos os eixos	0.64736717	0.43306069	0.38226889	0.32673834	0.27284124

Variância					
	Buffer=5 (uT)	Buffer=10 (uT)	Buffer=15 (uT)	Buffer=20 (uT)	Buffer=25 (uT)
Eixo X	1.28521979	0.72075546	0.51906485	0.36308264	0.26270425
Eixo Y	0.17720954	0.09830271	0.04632701	0.0291335	0.02689705
Eixo Z	0.57329657	0.17949053	0.14932602	0.12220666	0.07764496
Média de todos os eixos	0.6785753	0.33284957	0.2382393	0.17147427	0.12241542

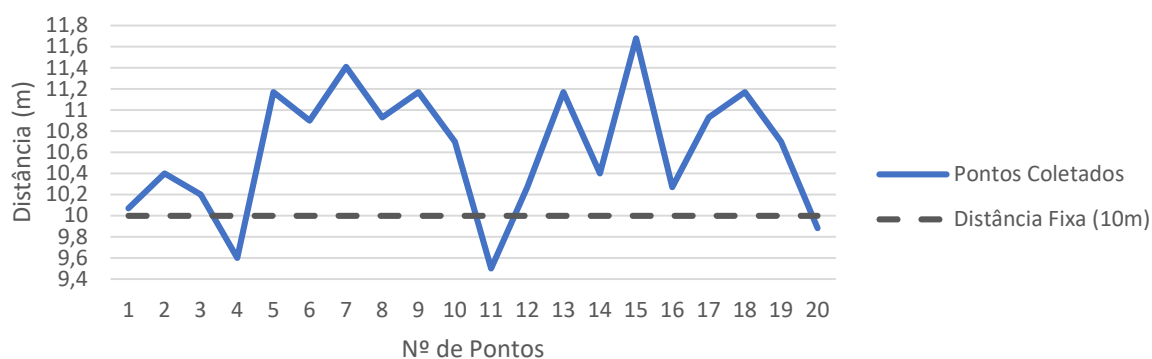
Anexo G



Variação da distância na marca de 9 metros



Variação da distância na marca de 10 metros



Variação da distância na marca de 12 metros

