



Instituto Superior de Engenharia

Politécnico de Coimbra

DEPARTAMENTO DE ENGENHARIA
ELETROTÉCNICA

Implementação de Navegação Autónoma baseada em Visão e Mapeamento 3D para Sistemas Robóticos Agrícolas

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

Especialização em Automação e Comunicações em Sistemas
Industriais

Autor

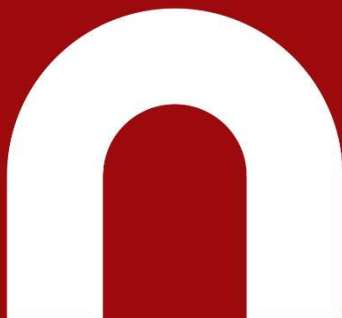
Joaquim Francisco Nascimento Ferreira

Orientador

Nuno Miguel Fonseca Ferreira

Co-Orientador

Inácio de Sousa Adelino da Fonseca



INSTITUTO POLITÉCNICO
DE COIMBRA

INSTITUTO SUPERIOR
DE ENGENHARIA
DE COIMBRA

Coimbra, dezembro de 2025

RESUMO

A automação de operações agrícolas exige soluções de robótica móvel capazes de navegar em ambientes exteriores pouco estruturados e dinâmicos.

Neste Trabalho de Projeto foi desenvolvido um sistema de navegação para implementar numa plataforma robótica com direção do tipo Ackermann, além deste sistema, efetuou-se uma avaliação para operar em vinhas, recorrendo ao ecossistema ROS 2 e ao framework de navegação NAV2.

Numa primeira fase efetuou-se o modelo do sistema robótico pretendido em URDF/SDF, integrando a cinemática de Ackermann, bem como diversos sensores para melhorar a perceção do ambiente e os atuadores necessários para a respetiva tarefa, o sistema foi desenvolvido no simulador Gazebo num cenário de teste, com vinha realista. Para a representação do meio ambiente utilizou-se a framework OctoMap, que permite criar mapas tridimensionais de ocupação a partir das nuvens de pontos recolhidos dos sensores de perceção. Sobre esta infraestrutura foi configurado o navegador, o planeador e o controlador de tarefas, respetivamente, através do navigation stack NAV2, do Smac Planner e do Vector Pursuit Controller. Foram incluídos os costmaps globais e locais adaptadas à geometria das linhas das videiras. A validação foi realizada através de vários ensaios de navegação em modo de simulação, incluindo trajetórias retilíneas entre cabeceiras, contornos completos de talhões de vinha e seguimento de sequências de waypoints. Os resultados obtidos evidenciam velocidades médias de operação e erros de posições inferiores a alguns centímetros, demonstrando a capacidade do sistema robótico para seguir percursos planeados de forma segura e precisa, permitindo assim uma solução robusta e flexível para navegação autónoma através da combinação da plataforma ROS 2 + NAV2 + OctoMap. A configuração pode ser reutilizada e estendida em futuros protótipos a desenvolver em ambiente real.

Palavras-chave: Robótica móvel, ROS 2, NAV2, OctoMap, direção Ackermann, navegação autónoma, vinhas.

ABSTRACT

The automation of agricultural operations requires mobile robotics solutions capable of navigating in unstructured and dynamic outdoor environments. In this project, a navigation system was developed to be implemented on a robotic platform with Ackermann steering. In addition to this system, an evaluation was carried out for operation in vineyards, using the ROS 2 ecosystem and the NAV2 navigation framework.

The model of the intended robotic system was created in URDF/SDF, integrating Ackermann kinematics, as well as various sensors to enhance environmental perception and the actuators required for the corresponding task. The system was developed in the Gazebo simulator within a test scenario featuring a realistic vineyard. For environmental representation, the OctoMap framework was used, which allows the creation of three-dimensional occupancy maps from point clouds collected by perception sensors. On this infrastructure, the navigator, planner, and task controller were configured using the NAV2 navigation stack, the Smac Planner, and the Vector Pursuit Controller, respectively. Global and local costmaps were included, adapted to the geometry of the vine rows.

Validation was carried out through several navigation trials in simulation mode, including straight trajectories between row ends, complete vineyard plot contours, and waypoint sequence following. The results obtained show average operating speeds and position errors of only a few centimeters, demonstrating the robotic system's capability to follow planned paths safely and accurately. This provides a robust and flexible solution for autonomous navigation through the combination of the ROS 2 + NAV2 + OctoMap platform. The configuration can be reused and extended in future prototypes to be developed in real environments.

Keywords: Mobile robotics, ROS 2, NAV2, OctoMap, Ackermann steering, autonomous navigation, vineyards

AGRADECIMENTOS

A realização desta dissertação só foi possível graças ao apoio, dedicação e incentivo de várias pessoas que, de diferentes formas, contribuíram para este trabalho.

Agradeço, em primeiro lugar, à minha namorada, pelo apoio incondicional, compreensão e paciência demonstrados ao longo de todo este percurso. A sua presença constante e motivação foram essenciais nos momentos mais exigentes.

À minha família, deixo um agradecimento profundo pelo suporte contínuo, pelos valores que me transmitiram e por acreditarem sempre nas minhas capacidades. Sem o vosso apoio, este caminho não teria sido possível.

Expresso também a minha sincera gratidão ao Engenheiro João Alves, pelo acompanhamento técnico, disponibilidade e orientação ao longo do desenvolvimento deste projeto. A sua experiência e contributos foram fundamentais para a evolução e qualidade deste trabalho.

Agradeço ao Professor Nuno Ferreira pela orientação, disponibilidade e contributos fundamentais ao longo desta dissertação. Agradeço também ao Professor Inácio Fonseca pelo apoio e pelas aprendizagens transmitidas durante o curso.

A todos, o meu muito obrigado.

ÍNDICE

RESUMO	i
ABSTRACT	iii
AGRADECIMENTOS.....	v
Índice.....	vii
Índice de figuras.....	ix
Índice de tabelas	xi
1 Introdução	1
1.1 Sistemas robóticos na agricultura.....	3
1.1.1 Processo da plantação.....	3
1.1.2 Processo da fertilização	4
1.1.3 Processo da colheita.....	6
1.1.4 Processo da poda e controlo de ervas daninhas	8
1.1.5 Controlo da plantação na agricultura	8
1.2 Resumo do capítulo	9
1.3 Objetivos do projeto.....	11
2 Secção de requisitos.....	13
2.1 Máquinas tradicionais	13
2.2 Tratores e Veículos com Lagartas	14
2.3 Cinemática do sistema robótico	16
2.3.1 Ackermann.....	16
2.4 Representação e orientação.....	19
2.4.1 Representação da Orientação.....	19
2.4.2 Representação da Posição	21
2.5 Controlador.....	23
2.5.1 Controladores clássicos	23
2.6 Sensores e localização	25
2.6.1 LiDAR	25
2.6.2 Inercial (IMU).....	27
2.6.3 Câmaras	28
2.7 Percepção e mapeamento.....	31
2.7.1 ROS.....	31

2.7.2	Odometria Visual	33
2.7.3	Visual SLAM.....	35
3	Simulador do sistema robótico.....	43
3.1	Desenho 3D.....	43
3.2	Ambiente de Simulação	47
3.3	Arquitetura de comunicação.....	49
4	Resultados do sistema robótico	51
4.1	Objetivos dos ensaios	51
4.2	Definição da experiência	51
4.3	Cenários de teste	54
4.4	Métricas de avaliação	56
4.5	Resultados de navegação	57
4.5.1	Seguimento de trajetória no cenário de referência	57
4.5.2	Navegação em torno da plantação	58
4.5.3	Navegação em toda a plantação	60
4.6	Resultados do mapeamento 3D com OctoMap.....	62
4.7	Discussão global.....	65
5	Conclusões e trabalhos futuros	70
	Referências	78

ÍNDICE DE FIGURAS

Figura 1-1 Casar [10]	4
Figura 1-2 Greenbot [12].....	4
Figura 1-3 Braço robótico [18]	5
Figura 1-4 Veículo Aéreo Não Tripulado [19].....	5
Figura 1-5 Sistema robótico que pulveriza remotamente [21]	6
Figura 1-6 Sistema robótico com rodas quando se desloca numa vinha [22].....	6
Figura 1-7 Análise dos morangueiros com redes neuronais [29].....	7
Figura 1-8 Robô Vegebot [30]	7
Figura 1-9 veículo Shrimp [42].....	9
Figura 2-1 Tratores.....	15
Figura 2-2 Trator movido a lagartas.....	15
Figura 2-3 Modelo de controlo preditivo	18
Figura 2-4 Filtro de Kalman Estendido.....	18
Figura 2-5 Definição da posição do sistema robótico num sistema referencial	22
Figura 2-6 Orientação do sistema robótico.....	22
Figura 2-7 Controlador PID	25
Figura 2-8 LiDAR [68].....	27
Figura 2-9 IMU [127]	28
Figura 2-10 FrameWork ROS2 [131].....	32
Figura 3-1 Desenho 3D do sistema robótico.....	44
Figura 3-2 Desenho 3D do sistema robótico com um angulo de -60°	44
Figura 3-3 Corpos rígidos do sistema robótico	45
Figura 3-4 Terreno desenvolvido para a simulação.....	45
Figura 3-5 Imagem da câmara RGB-D.....	46
Figura 3-6 Imagem do sistema robótico e do ambiente virtual.....	48
Figura 4-1 Ambiente de simulação.....	52
Figura 4-2 Trajetória retilínea.....	54
Figura 4-3 Trajetória em torno do cenário	55
Figura 4-4 Trajetória no cenário	56
Figura 4-5 Trajetória retilínea.....	58

Figura 4-6 Trajetória em torno do cenário.....	59
Figura 4-7 Trajetória de toda a vinha.....	60
Figura 4-8 OctoMap na trajetória retilínea (simulação n.º 1).....	63
Figura 4-9 OctoMap na trajetória em torno do cenário (simulação n.º 2)	63
Figura 4-10 OctoMap na trajetória no cenário (simulação n.º 3)	64

ÍNDICE DE TABELAS

Tabela 1-1: Sistemas robóticos existentes.....	10
Tabela 2-1: Características dos LiDAR.....	27
Tabela 2-2: Tipos de câmaras.....	30
Tabela 3-1: Parametrização do simulador.....	50
Tabela 4-1: Configuração NAV2.....	53
Tabela 4-2: Dados simulação n.º 1.....	58
Tabela 4-3: Dados simulação n.º 2.....	59
Tabela 4-4: Dados simulação n.º 3.....	61
Tabela 4-5: Dados visuais do OctoMap.....	62

1 INTRODUÇÃO

A robótica tem desempenhado um papel importante na modernização de diversos setores, da indústria até à agricultura, promovendo inovações que visam o aumento da eficiência, a redução de custos e a minimização dos impactos ambientais [1, 2].

O setor agrícola está a passar por uma mudança, impulsionada pela integração de tecnologias, tais como a inteligência artificial, que estão a permitir a automação de processos que anteriormente dependiam do trabalho humano [3,4].

Esta transformação é impulsionada por dois fatores, sendo o primeiro a necessidade de desenvolver métodos de produção mais sustentáveis, face ao aumento da procura de alimentos e o segundo fator relacionado com os diversos desafios, incluindo a escassez de trabalhadores agrícolas e os efeitos das alterações climáticas [5] entre outros.

A evolução da mecanização a tem desenvolvido um papel crucial na modernização deste setor, substituindo os processos manuais por máquinas mais eficientes [6].

No entanto, o domínio da robótica evoluiu para além dos limites da mecanização clássica, dando lugar a sistemas inteligentes capazes de tomar decisões em tempo real e adaptar-se às condições ambientais para executar tarefas com precisão. Para além da automação das operações agrícolas tradicionais, a instalação de uma variedade de sensores permite não só efetuar uma análise dos dados, mas também o desenvolvimento de sistemas de previsão baseados em inteligência artificial.

A inclusão de sensores inteligentes e redes de comunicação permitem recolher e processar diferentes informações de forma pormenorizada sobre o solo, o clima e o crescimento das plantas, permitindo assim ajustes em tempo real para maximizar a produtividade de forma eficiente e sustentável [7].

Esta abordagem, também conhecida como agricultura de precisão, tornou-se essencial para uma gestão sustentável das explorações agrícolas, reduzindo o desperdício e promovendo a utilização eficiente dos fatores de produção agrícola como a fertilização e a colheita de alimentos.

Outro avanço da tecnologia é a utilização de múltiplos robôs a executar tarefas em simultâneo, tarefas distintas tais como a plantação, a poda e a colheita. Este avanço permite reduzir a dependência de uma só máquina de grandes dimensões e assim os agricultores passam a contar com vários robôs de dimensões mais reduzidas e de baixo custo a desenvolver tarefas em simultâneo, facilitando assim o trabalho [3], ou até ter um robô de baixo custo que consiga efetuar tarefas diversas.

Esta inovação não só melhora a eficiência, como também reduz a compactação do solo e o impacto ambiental da maquinaria agrícola pesada. No entanto, a aplicação da robótica no setor agrícola ainda enfrenta desafios, incluindo custos excessivos, a necessidade de formação técnica e a adaptação das infraestruturas rurais para apoiar estas tecnologias modernas.

Apesar dos grandes desafios, o investimento contínuo no desenvolvimento e na investigação tem levado à criação de soluções mais acessíveis e adaptadas às necessidades dos produtores, permitindo assim evoluir para sistemas agrícolas mais automatizados e interligados.

Neste contexto, pretende-se explorar o papel da robótica na agricultura, analisando os seus avanços tecnológicos, aplicações práticas, desafios e implicações para o futuro deste setor.

1.1 Sistemas robóticos na agricultura

Os avanços da robótica móvel, da visão computacional e da inteligência artificial tem impulsionado uma transformação significativa na agricultura moderna. Estas tecnologias permitem automatizar operações manuais como a plantação, a fertilização, a colheita e a poda, bem como reduzir os custos operacionais, aumentar a precisão na aplicação de insumos e minimizar o impacto ambiental [8].

A integração de sensores inteligentes e algoritmos de percepção aumenta a capacidade de detecção precoce de doenças ou anomalias nas culturas, promovendo intervenções rápidas e eficazes. A aplicação de sistemas robóticos permite otimizar recursos e contribui para uma agricultura mais sustentável e tecnologicamente avançada.

1.1.1 Processo da plantação

A automação da plantação surge como resposta ao aumento da procura de alimentos e à necessidade de melhorar a eficiência dos sistemas produtivos [9]. A agricultura de precisão, baseada na aplicação localizada de sementes e recursos naturais, tem demonstrado reduzir desperdícios e aumentar a produtividade [10].

Robôs agrícolas como o Cäsar, apresentado na Figura 1-1, utilizam tecnologia de posicionamento de elevada precisão baseada em sistemas de navegação por satélite e técnicas de correção de posição em tempo real que aumenta a precisão do GNSS RTK-GNSS, (GNSS, Global Navigation Satellite System), (RTK, Real-Time Kinematic), permitindo precisões de cerca de 3 cm durante a navegação [12], capaz de efetuar a deposição rigorosa das sementes.



Figura 1-1 Casar [10]

Outro exemplo é o Greenbot apresentado na Figura 1-2, um trator autónomo 4WS capaz de operar durante longos períodos e detetar obstáculos de forma automática [11]. Estes sistemas reduzem a variabilidade da plantação, melhoram a uniformidade das culturas e permitem operações contínuas sem necessidade de pausas humanas [12]. Os principais exemplos encontram-se sintetizados na Tabela 1.1.



Figura 1-2 Greenbot [12]

1.1.2 Processo da fertilização

Na fertilização utilizam-se tecnologias com GPS, SIG (Sistema de Informação Geográfica) e outros sensores avançados para ajustar as doses de nutrientes aplicadas às necessidades específicas dos solos [13-15]. Tecnologias como a Smart N. Fertilisation permitem a aplicação localizada de azoto, reduzindo impactos como a

diminuição da qualidade do azeite causada pelo excesso de azoto [16] ou por exemplo os efeitos negativos do fósforo na produção de tomate [17].

Exemplos de manipuladores robóticos em estufas, sistemas estes equipados com câmaras RGB e sensores laser, que permitem detetar doenças, tais como o oídio e o vírus da murchidão dos tomateiros com taxas de precisão de 64% a 90%, respetivamente [18] (Figura 1-3).

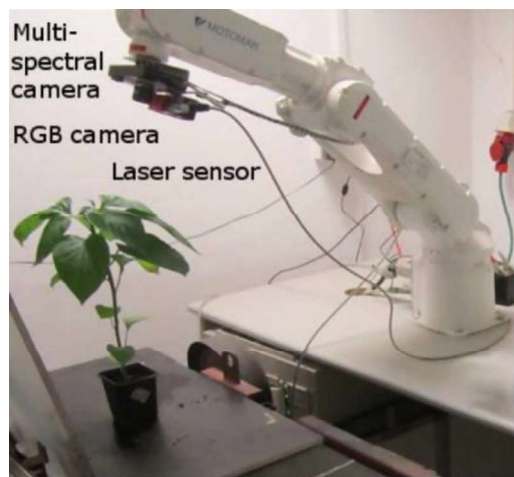


Figura 1-3 Braço robótico [18]

Outro tipo de sistemas robóticos, Veículos Aéreos Não Tripulados (VANTs) são equipados com câmaras térmicas ou multiespectrais que permitem calcular o crescimento da vegetação, o Normalized Difference Vegetation Index (NDVI) para monitorizar o estado das culturas [19-20], ou Quadricópteros de pulverização permitem reduzir a quantidade de químicos utilizados [21] (Figura 1-4 e 1-5).

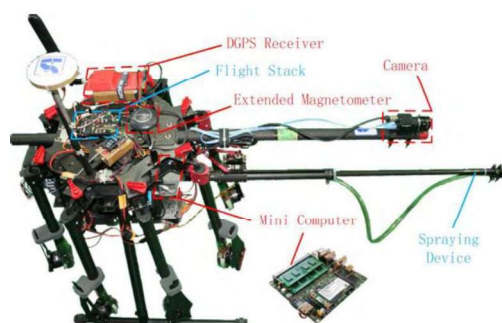


Figura 1-4 Veículo Aéreo Não Tripulado [19]



Figura 1-5 Sistema robótico que pulveriza remotamente [21]

Nas vinhas, os robôs móveis conseguem reduzir a utilização de pesticidas até 45% através de algoritmos de detecção de folhagem e das uvas [22] (Figura 1-6).



Figura 1-6 Sistema robótico com rodas quando se desloca numa vinha [22]

1.1.3 Processo da colheita

No processo de colheita é necessário um sistema de visão computacional, sensores multiespectrais para identificar e manipuladores robóticos para remover frutos maduros com elevada precisão [23-24]. Estes sistemas aceleram o processo, reduzem danos e melhoram a qualidade do produto pós-colheita [25-27].

Contudo, o custo inicial elevado e a variabilidade ambiental constituem desafios à sua adoção [28]. Em culturas mais complexas, como videiras e pimenteiros são necessários sensores mais robustos assim como algoritmos avançados, incluindo redes neurais convolucionais (CNN) e técnicas de segmentação multiespectral, para

alcançar níveis aceitáveis de desempenho. Na colheita dos morangos Fernández [29] mostrou que um robô manipulador com uma câmara RGB-D e com um algoritmo baseado em redes neurais R-CNN obteve 74,1% de sucesso, com $F1 = 0,9$, apresentando um equilíbrio entre precisão e o recall, ou seja, o algoritmo deteta bem os morangos e erra pouco (Figura 1-7).

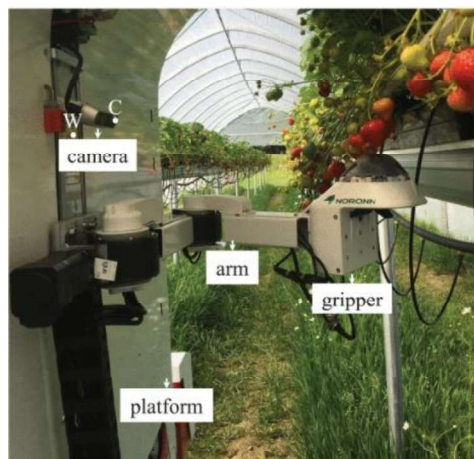


Figura 1-7 Análise dos morangueiros com redes neurais [29]

O sistema robótico Vegebot, conforme se mostra na Figura 1-8, foi desenvolvido para colheita de alfaces e registou 91% de sucesso na localização e 82% de precisão e classificação [30-31].

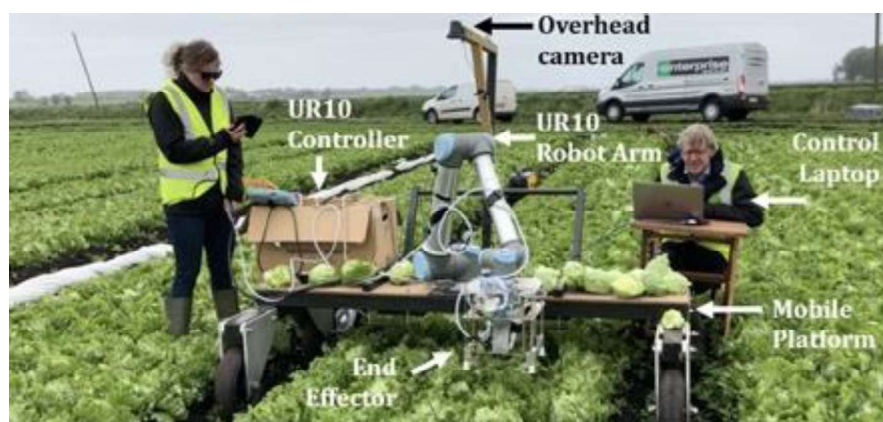


Figura 1-8 Robô Vegebot [30]

1.1.4 Processo da poda e controlo de ervas daninhas

Na poda e no controlo de ervas daninhas recorrem a câmaras RGB, sensores de profundidade e algoritmos de segmentação aliados a IA para distinguir culturas de infestantes [32-33]. Métodos como a Transformada de Hough e a aprendizagem profunda permitem guiar os robôs através de linhas de cultivo com elevada fiabilidade [34]. Estes sistemas operam continuamente, reduzem a necessidade de herbicidas e diminuem custos operacionais a longo prazo [35-36].

Contudo, permanecem desafios como a adaptação a terrenos irregulares e a necessidade de calibração frequente [37]. Espera-se que o avanço da IA e dos sensores melhore significativamente a eficácia destas soluções, tornando a poda robotizada uma ferramenta essencial na agricultura sustentável [38].

1.1.5 Controlo da plantação na agricultura

O controlo da plantação beneficia da integração de sensores IoT, IA e plataformas robóticas capazes de recolher dados ambientais e prever o rendimento agrícola [34,39]. Sensores de humidade, de temperatura e de nutrientes transmitem informação em tempo real, auxiliando o sistema de controlo de irrigação e fertilização de forma inteligente [40]. A utilização de VANTs, também designados como drones, com sensores multiespectrais podem permitir a identificação do stress hídrico, nutricional bem como a evolução das pragas na plantação [41].

Plataformas robóticas como a Shrimp (Figura 1-9) utilizam a tecnologia GPS, e sistemas de navegação inercial (INS) assim como redes neuronais do tipo perceção multicamadas (MLP) e Rede Neural Convolutiva (CNN), alcançando taxas de deteção de maçãs de 82,5% e $R^2 = 0,826$ na estimativa de rendimento [42]. Em ambientes desafiantes, como as vinhas do Douro, o sistema robótico Agrob V14 combina várias tecnologias, câmaras RGB, IR, LIDAR assim como etiquetas RFID

com o VineSLAM para poder navegar autonomamente mesmo com o sistema GNSS degradado, superando obstáculos e declives até 30% [43].



Figura 1-9 veículo Shrimp [42]

1.2 Resumo do capítulo

O estado da arte demonstra que a integração dos sistemas robóticos com a IA permite que os sistemas autônomos possam constituir um pilar central para o futuro da agricultura de precisão. Estes sistemas melhoram a produtividade, reduzem custos e promovem práticas agrícolas ambientalmente sustentáveis. A Tabela 1-1 sintetiza os principais robôs, técnicas e resultados apresentados, fornecendo uma visão clara do estado da arte que fundamenta o seu desenvolvimento.

Tabela 1-1: Sistemas robóticos existentes.

Processo agrícola	Sistema Exemplos:	Tipo de plataforma	Tecnologias principais	Função Aplicação	Resultados principais
Plantação fertilização	Cäsar	Robô agrícola terrestre	RTK-GNSS de alta precisão	Fertilização e navegação entre linhas	Precisão ~3 cm; redução de desperdício
Plantação operações repetitivas	Greenbot	Trator autônomo 4WS	Sensores de obstáculos, operação contínua	Sementeira, lavoura e fertilização	Operação 10 h/dia; elevada capacidade de carga
Fertilização saúde das plantas	Robô de estufa (manipulador 6-DOF)	Base móvel + manipulador	Câmara RGB, sensor laser	Deteção de doenças (oídio e vírus de murchidão)	64% (oídio) e 90% (vírus) de precisão
Monitorização e irrigação	VANTs multiespectrais	Drone	Câmaras térmicas e multiespectrais, NDVI	Monitorização das culturas e irrigação de precisão	Mapeamento de alta resolução; deteção de stress hídrico
Pulverização localizada	Quadricóptero pulverizador	Drone multirrotor	Depósito de ureia/pesticida	Aplicação localizada de químicos	Redução de químicos; menor risco humano
Pulverização em vinhas	Robô terrestre para vinha	Robô com rodas	RGB + sensores de distância + FDA/GDA	Pulverização controlada	Redução até 45% do uso de pesticidas
Colheita de morangos	Robô de colheita	Robô móvel com manipulador	RGB-D + R-CNN	Deteção e colheita seletiva	74,1% sucesso; F1 = 0.9
Colheita de alface	Vegebot	Robô móvel + 6-DOF	Duas câmaras RGB + R-CNN	Colheita autônoma	91% localização; 82% classificação
Estimativa de rendimento	Plataforma Shrimp	Veículo terrestre	GPS, INS, CNN, WS, CHT	Estimativa de produção em pomares	Deteção 82,5%; $R^2 = 0.826$

1.3 Objetivos do projeto

O presente Trabalho de Projeto tem como objetivo geral desenvolver e validar uma arquitetura de navegação autónoma para um sistema robótico agrícola com direção do tipo Ackermann, recorrendo ao ecossistema ROS 2 e à framework NAV2, de modo a permitir a sua futura aplicação em ambientes agrícolas estruturados, nomeadamente em vinhas.

De forma mais específica, pretende-se modelar o sistema robótico em URDF/SDF, integrando a sua cinemática, os atuadores e os sensores principais; desenvolver um ambiente de simulação em Gazebo representativo de um cenário agrícola; integrar sensores de perceção, designadamente uma IMU e uma câmara RGB-D, para suporte à localização e ao mapeamento; implementar o mapeamento tridimensional do ambiente com recurso ao OctoMap; configurar os módulos de planeamento e controlo da navegação no NAV2, incluindo o Smac Planner e o Vector Pursuit Controller; e, por fim, avaliar o desempenho do sistema em diferentes cenários de navegação, com base em métricas como o erro de trajetória, o tempo de missão, a percentagem de waypoints atingidos e a coerência do mapa tridimensional gerado.

Com este trabalho pretende-se demonstrar a viabilidade técnica de uma solução modular, robusta e extensível para navegação autónoma em contexto agrícola, constituindo uma base para futura implementação num protótipo físico.

2 SECÇÃO DE REQUISITOS

Pretende-se neste capítulo apresentar o processo de conceção do sistema robótico, articulando de forma estruturada os fundamentos teóricos, bem como as opções de engenharia adotada. A construção da arquitetura é central para validar, através do simulador ou através de ensaios experimentais, a viabilidade das soluções propostas.

Inicia-se com o enquadramento das máquinas agrícolas tradicionais e da sua evolução tecnológica, justificando assim a necessidade de veículos autónomos mais leves, modulares e energeticamente eficientes. A seguir, procede-se à análise comparativa dos principais modelos de locomoção utilizados em robótica móvel: a) diferencial e b) Ackermann, destacando as implicações cinemáticas de cada solução para a navegação em ambientes agrícolas.

Complementarmente, descrevem-se os métodos de estimação da pose, e os sistemas de controlo adotados, e mostra-se a integração dos sensores essenciais para o funcionamento do sistema robótico, incluindo um IMU, câmaras RGB-D, ultrassons e a utilização de frameworks avançadas tais como o OctoMap e ROS 2.

A secção final, aborda o processo de montagem e validação do protótipo, evidenciando os desafios enfrentados bem como as estratégias implementadas para garantir uma operação estável.

2.1 Máquinas tradicionais

O setor agrícola tem sido historicamente apoiado por máquinas de grande porte, destacando-se os tratores, as ceifeiras e os pulverizadores, que permitiram uma revolução produtiva ao longo do século XX [44]. Contudo, estas plataformas apresentam diversas limitações, tais como produção em massa, ou o elevado consumo energético, a baixa modularidade e a escassez de sensores embutidos

[45,46]. Tipicamente este tipo de veículos, apresentam um comportamento cinemático baseado na direção do tipo Ackermann, que é o mais adequado para terrenos amplos e desobstruídos, revelando-se, no entanto, pouco eficiente em contextos de elevada densidade vegetal, ou declives ou até em manobras de precisão [47].

A crescente necessidade de operações autónomas, devido à falta de trabalhadores, ou até por operações contínuas, tais como a recolha inteligente de dados, fomentou o desenvolvimento de veículos robóticos mais compactos. Estes sistemas robóticos proporcionam maior agilidade, menor impacto no solo e capacidade de integração de sensores avançados, tornando-se candidatos ideais para ambientes agrícolas modernos.

2.2 Tratores e Veículos com Lagartas

Os tratores constituem o eixo central da mecanização agrícola, combinando potência, tração e adaptabilidade (Figura 2-1) [48]. Contudo, o seu peso significativo contribui para a compactação do solo, um dos problemas para a produtividade agrícola [49].

A digitalização recente introduziu a tecnologia de posicionamento com GPS, a telemetria e os sistemas de gestão agrícola, aproximando estes veículos do paradigma da automação [50]. Paralelamente, surgem propostas de tratores autónomos, eliminando cabines e integrando sensores avançados. Porém, as barreiras económicas e regulatórias limitam a sua adoção massiva [51].

Adicionalmente, têm vindo a ser amplamente adotados sistemas com guiadores automáticos, em que o trator segue trajetórias pré-definidas com base em GNSS/GPS (frequentemente com correção RTK), reduzindo a intervenção contínua do operador. Estes sistemas aumentam a precisão, diminuem

sobreposições e melhoram a eficiência operacional, sendo uma etapa intermédia para a automação total.



Figura 2-1 Tratores

Os tratores de lagartas apresentam vantagens claras em terrenos lamacentos, inclinados ou de baixa capacidade de suporte, distribuindo melhor o peso e aumentando a tração (Figura 2-2) [52]. Apesar disso, o custo de manutenção e a maior complexidade mecânica restringem a sua utilização a explorações de maior escala [53]. Esta análise contextualiza a decisão de optar por uma plataforma robótica mais leve, baseada em cinemática Ackermann, capaz de operar entre fileiras estreitas sem comprometer o solo.



Figura 2-2 Trator movido a lagartas

2.3 Cinemática do sistema robótico

O desenvolvimento de veículos autônomos para aplicações agrícolas tem evoluído rapidamente, com destaque para os tratores, amplamente utilizados na agricultura devido à sua robustez e versatilidade. Para a implementação de sistemas robóticos agrícolas autônomos, foi escolhido o modelo de trator com base na sua capacidade de operar em terrenos irregulares e em condições ambientais adversas.

Além disso, os tratores oferecem a vantagem de um sistema de tração eficiente, essencial para a movimentação e execução de diferentes tarefas de forma eficaz, tais como a semeadura, a colheita e a fertilização.

Contudo, para o funcionamento autônomo é necessário considerar os sistemas de controlo da mobilidade sendo um dos componentes mais importantes a cinemática da estrutura robótica, fundamental para determinar o movimento.

Dentro desse contexto, existem dois tipos de estruturas amplamente utilizados para veículos móveis autônomos: a cinemática diferencial e de Ackermann.

2.3.1 Ackermann

O sistema de direção Ackermann é utilizado em veículos terrestres, devido à sua capacidade de minimizar o deslizamento lateral dos pneus durante as curvas, garantindo assim uma trajetória mais eficiente e estável. Este mecanismo é essencial para a mobilidade de veículos autônomos, máquinas agrícolas e claro os robôs móveis, até porque permite uma movimentação mais precisa em ambientes estruturados e não estruturados [54,55].

O principal objetivo é garantir que, ao realizar uma curva, todas as rodas do veículo sigam trajetórias circulares concêntricas. Esta configuração é obtida através da diferenciação no ângulo de estiramento das rodas dianteiras: a roda interna da curva roda em num ângulo maior do que a roda externa. Este método reduz o escorregamento lateral dos pneus, aumentando assim a estabilidade e reduzindo o seu desgaste [56,57].

Para descrever a cinemática, utiliza-se um modelo simplificado conhecido como o modelo da bicicleta, no qual o veículo é representado por um único eixo dianteiro e um único eixo traseiro. A posição do veículo no espaço é descrita pelas suas coordenadas cartesianas (x, y) e pelo ângulo de orientação θ .

Os comandos de controlo são a velocidade linear v e o ângulo de estiramento δ das rodas dianteiras. A cinemática do sistema é dada pelas seguintes equações [58]:

$$\dot{x} = v * \cos(\theta) \quad (1)$$

$$\dot{y} = v * \sin(\theta) \quad (2)$$

$$\dot{\theta} = \frac{v}{L} * \tan(\delta) \quad (3)$$

Onde L representa a distância entre os eixos dianteiro e o traseiro do veículo. Estas equações assumem que não há deslizamento lateral, sendo uma boa aproximação para velocidades moderadas e superfícies regulares [59].

A utilização da cinemática do tipo Ackermann está destinada a uma grande variedade de aplicações, tais como os veículos autónomos do tipo máquinas agrícolas. No setor de agricultura de precisão, os veículos equipados com direção Ackermann são usados para navegação em campos de cultivo, permitindo tarefas como colheita automatizada, de pulverização seletiva e de monitoramento de lavouras [56][60].

Em ambientes urbanos, a geometria Ackermann é essencial para garantir que as trajetórias planeadas sejam seguidas com precisão, evitando obstáculos e otimizando a eficiência energética do deslocamento. Para isso, estratégias de controlo avançadas, como o modelo de controlo preditivo (MPC) (Figura 2-3), são amplamente utilizados, garantindo um desempenho otimizado levando em conta as restrições cinemáticas do veículo [61].

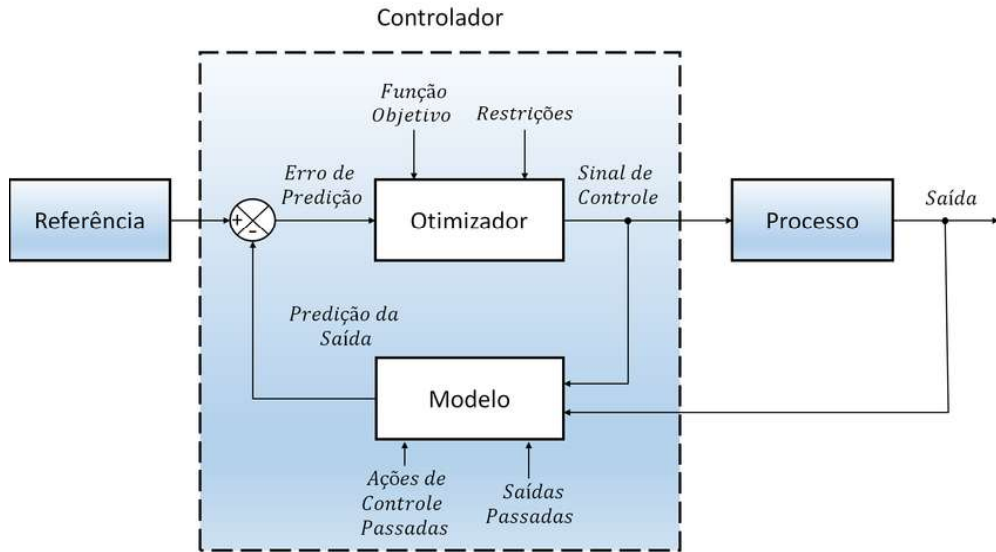


Figura 2-3 Modelo de controle preditivo

Além das restrições, para melhorar a pose do veículo durante a navegação, técnicas como o Filtro de Kalman Estendido (EKF) são amplamente aplicadas, (Figura 2-4) para obter a pose do sistema.

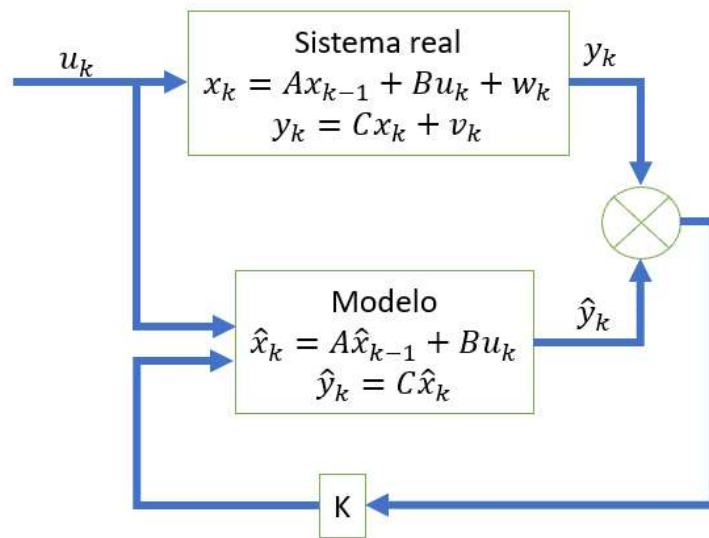


Figura 2-4 Filtro de Kalman Estendido

Estas técnicas combinam informações dos sensores como o GPS, as Unidades de Medição Inercial (IMUs) e os encoders internos para estimar a posição e a orientação, ou seja a pose do veículo em tempo real, minimizando os erros acumulados ao longo do deslocamento [62].

Apesar da sua eficiência para navegação em terrenos regulares, a cinemática Ackermann apresenta desafios quando aplicada em superfícies irregulares ou em velocidades elevadas. Nestas condições, efeitos como deslizamento lateral dos pneus e forças inerciais tornam-se mais significativos, exigindo modelos mais complexos para descrever o comportamento do veículo [63].

Para mitigar as limitações referidas, abordagens híbridas que combinam modelos cinemáticos e dinâmicos estão a ser exploradas como a combinação da cinemática Ackermann com a cinemática diferencial. Outras estratégias de controlo podem ser baseadas em Deep Learning com intuito de melhorar a precisão da navegação em ambientes não estruturados.

Estas abordagens procuram aumentar a robustez e adaptabilidade do sistema em situações dinâmicas e desafiadoras [64].

2.4 Representação e orientação

Uma abordagem comum para descrever a posição e a orientação de um sistema robótico é utilizar um sistema de coordenadas tridimensional, onde os eixos X , Y e Z representam as direções espaciais. A orientação do sistema robótico pode ser descrita por meio de ângulos de Euler ou quatérniões, que indicam a rotação em torno desses eixos.

2.4.1 Representação da Orientação

A rotação em torno de um eixo também pode ser representada por quatérniões, que possuem duas componentes principais: uma escalar, denotada por q_s , e uma vetorial, qv conforme se apresenta na equação 4:

$$q = (q_s, q_v) = q_4 + iq_1 + jq_2 + kq_3 \quad (4)$$

Onde i, j e k são componentes da parte vetorial $q_v = (q_1, q_2, q_3)$ e q_4 é a parte escalar $q_s = q_4$.

A partir de uma rotação em torno de um eixo, é possível calcular as componentes do quaternião através das seguintes equações:

$$q_v = \sin\left(\frac{\Lambda}{2}\right) \begin{bmatrix} k_x \\ k_y \\ k_z \end{bmatrix} \quad (5)$$

$$q_s = \cos\left(\frac{\Lambda}{2}\right) \quad (6)$$

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. É importante destacar que um quaternião possui uma norma unitária, o que possibilita interpretar um ponto numa esfera unitária de quatro dimensões, conforme está representado na expressão seguinte:

$$\|q\| = \sqrt{q_w^2 + q_x^2 + q_y^2 + q_z^2} \quad (7)$$

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

$$R(q) = \left(q_s^2 - \|q_v\|^2\right) I_3 + 2q_v q_v^T + 2q_s (q_v) \quad (8)$$

Analisando a expressão 8, permite obter a expressão do quaternião através dos elementos da matriz de rotação da seguinte forma:

$$q_v = \frac{1}{4q_s} \begin{bmatrix} r_{23} - r_{32} \\ r_{31} - r_{13} \\ r_{12} - r_{21} \end{bmatrix} \quad (8)$$

$$q_s = \frac{1}{2} \sqrt{1 + r_{11} + r_{22} + r_{33}} \quad (9)$$

A representação por quatérnios é a mais vantajosa pelo facto de não ser afetada por singularidades e apresentar mais eficiência a nível computacional [65].

2.4.2 Representação da Posição

Um sistema robótico dentro de um ambiente físico e as suas ações, estão intrinsecamente ligadas à sua localização precisa no mapa, incluindo tanto a sua posição como a sua orientação.

Observando a Figura 2-5, definimos um sistema de coordenadas num referencial XY como sendo o ponto inicial da navegação do sistema robótico.

Supondo que este está no ponto de origem do referencial, com um novo sistema de coordenadas R. Desta forma o vetor r representa o deslocamento entre os sistemas de coordenadas W e o R, presentes no ponto P.

Este sistema de coordenadas tanto pode ser definido em coordenadas polares como em coordenadas cartesianas, conforme se apresenta na Figura 2-6.

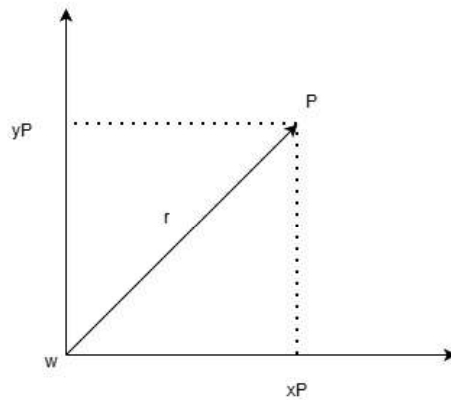


Figura 2-5 Definição da posição do sistema robótico num sistema referencial

A posição do sistema robótico em relação a W é descrita pela posição e orientação do sistema de coordenadas local R . O sistema robótico do tipo Ackermann, representado na Figura 2-6 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.
3. O ponto O representa o ponto-alvo no espaço, estimado pela triangulação da interseção das linhas de visão definidas pelos ângulos α e β .

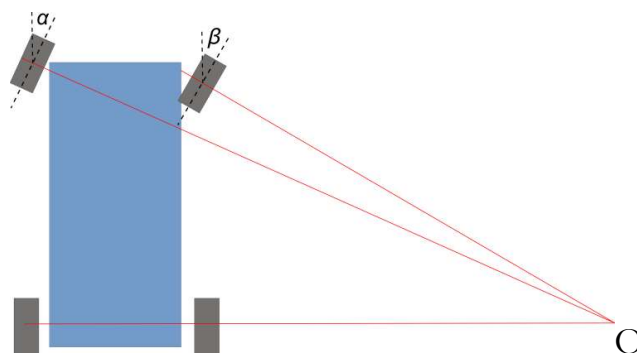


Figura 2-6 Orientação do sistema robótico.

2.5 Controlador

Em sistemas autónomos, é indispensável a existência de uma componente computacional que permita a execução das tarefas previstas. O controlador de um sistema robótico atua como o “processador” de toda a informação adquirida pelos sensores e é responsável por criar as ações necessárias para o movimento do sistema robótico.

O sistema de controlo adquire os dados provenientes dos sensores, interpreta-os e determina a ação de controlo mais adequada, enviando posteriormente os comandos para os atuadores. Sem esta camada de controlo, o sistema robótico não conseguiria adaptar-se ao ambiente envolvente, nem reagir a alterações nas condições de operação.

Dependendo da aplicação, podem ser adotadas diferentes estratégias de controlo, nomeadamente em malha aberta ou em malha fechada. Num sistema de malha aberta, os atuadores são comandados diretamente pelo utilizador ou por um sinal pré-definido, sem realimentação do estado atual. Num sistema de malha fechada, existe uma realimentação contínua das variáveis de interesse, permitindo ao controlador ajustar o seu comportamento em função do erro entre a referência e a variável a controlar.

A estratégia de controlo mais adequada depende, das exigências de desempenho e robustez de cada aplicação. Entre as várias abordagens existentes, os controladores clássicos do tipo Proporcional–Integral–Derivativo (PID) que se destacam pela sua ampla utilização em sistemas robóticos e em engenharia em geral [66].

2.5.1 Controladores clássicos

Os controladores clássicos PID, a ação de controlo resulta da combinação de três componentes fundamentais: proporcional, integral e derivativa. A componente proporcional (P) atua sobre o erro instantâneo entre o valor de referência e o valor

medido, permitindo uma correção imediata. A componente integral (I) acumula o erro ao longo do tempo, contribuindo para eliminar o erro em regime permanente.

A componente derivativa (D) reage à variação temporal do erro, antecipando a sua evolução e ajudando a amortecer oscilações. A saída do controlador PID é a soma ponderada das três componentes, podendo ser expressa pela seguinte equação geral:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (19)$$

No sistema robótico com direção Ackermann, é essencial efetuar um controle preciso para garantir a trajetória de referência. Para tal, recorre-se a dois controladores distintos que atuam em cascata, um dedicado ao controle da velocidade e outro ao controle da direção.

O primeiro controlador clássico controla a velocidade do sistema robótico, ajustando dinamicamente o sinal de comando (por exemplo, o PWM) aplicado aos motores das rodas traseiras. O termo proporcional corrige os desvios em relação à velocidade de referência, o termo integral compensa erros acumulados ao longo do tempo, assegurando um movimento mais uniforme, e o termo derivativo atenua variações bruscas, reduzindo oscilações indesejadas.

O segundo controlador clássico atua sobre o ângulo de estiramento das rodas dianteiras, garantindo que o sistema robótico siga a trajetória desejada com precisão.

Desta forma, compensa os pequenos desvios de orientação e contribui assim para um movimento suave e previsível, evitando oscilações excessivas na direção e melhorando a estabilidade global do sistema.

Em conjunto, estes dois controladores permitem implementar comandos de movimentação baseados na distância a percorrer e na direção desejada, assegurando

um deslocamento autónomo, eficiente e seguro em cenários com trajetórias potencialmente complexas. Assim, o modelo de Ackermann é explorado de forma eficaz em aplicações industriais, agrícolas e em veículos autónomos(Figura 2-7) [67].

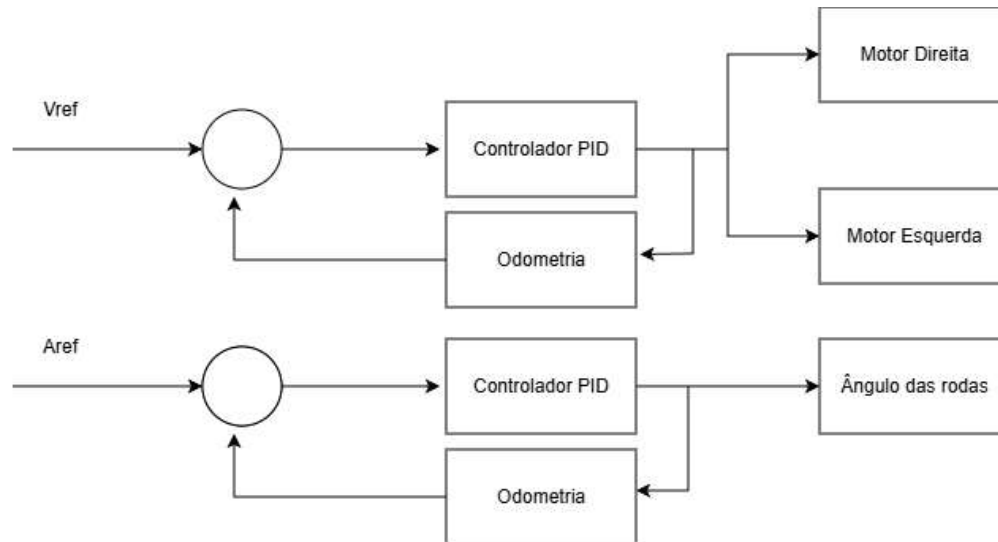


Figura 2-7 Controlador PID

2.6 Sensores e localização

2.6.1 LiDAR

LiDAR, sigla utilizada para Light Detection and Ranging, é um termo que tem sido utilizado para designar esta nova tecnologia de sensoriamento remoto. O termo lidar (Laser Detection and Ranging) também é utilizado, porém, com mais frequência em aplicações militares. Foram Schawlow & Townes (1958), pela primeira vez, que estabeleceram a teoria para obter a amplificação de luz por emissão estimulada de radiação (laser) com base nos fundamentos estabelecidos por Einstein em 1917.

Nas décadas de 1960 e 1970, várias experiências demonstraram o poder de utilização do laser em sensoriamento remoto, incluindo diversas aplicações, como monitoramento atmosférico e estudos oceanográficos. Durante a década de 1980, o laser passou a ser utilizado na medição de distâncias, sendo implementados instrumentos como o NASA's Atmospheric Oceanographic Lidar (AOL) e Airborne

Topographic Mapper (ATM). Mais recentemente, na década de 1990, foram desenvolvidos sensores mais confiáveis, como o Shuttle Laser Altimeter (SLA) e o Mars Observer Laser Altimeter (MOLA) usados para medidas em extensas áreas da superfície terrestre e dos oceanos.

O princípio de funcionamento do sistema de varredura laser consiste na emissão de um pulso laser de uma plataforma (aérea, terrestre ou orbital) com uma elevada frequência de repetição. O tempo de retorno dos pulsos laser entre a plataforma e os alvos é medido pelo sensor, permitindo a estimativa destas.

O sistema lidar apresenta a sua própria fonte de energia, neste caso, uma fonte de luz, o laser, a qual emite radiação eletromagnética em ondas curtas (1-10 μm) sendo sensível às perturbações atmosféricas. Por esse motivo, é considerado um método direto na captura de dados e classificado como um sensor ativo. Uma vantagem dos sensores de varredura a laser em relação aos clássicos sensores passivos (fotografias aéreas e imagens de satélite) é que os scanners a laser não dependem do sol como uma fonte de iluminação. Sendo assim, a análise dos dados não é prejudicada por sombras causadas por nuvens ou objetos vizinhos.

Os sistemas ALS (Airborne Laser Scanner) são compostos por um sensor laser que essencialmente regista o tempo entre o sinal emitido e recebido de um determinado ponto, um Sistema Inercial de Navegação (INS – Inertial Navigation System) e um recetor GPS numa plataforma aerotransportada. Em terra, é necessária uma rede de GPS que trabalhe de forma conjunta ao GPS instalado na plataforma. Princípios e componentes de um sistema ALS. O conjunto total de informações é armazenado e processado, resultando em pontos com extrema precisão (x , y e z), que correspondem a superfícies e objetos presentes no terreno, conforme representado na Figura 2-9 [68].



Figura 2-8 LiDAR [68]

Tabela 2-1: Características dos LiDAR.

Tipo de LiDAR / Característica	Descrição	Vantagens	Desvantagens	Aplicações típicas
ALS — Airborne Laser Scanning	Laser montado em avião ou drones; mede distâncias ao solo e estruturas.	Grande cobertura; elevada precisão; ideal para mapeamentos extensos.	Custo muito elevado; dependente de INS e GPS.	Topografia agrícola, cartografia 3D, planeamento territorial.
LiDAR Terrestre (TLS)	Sensor fixo ou num veículo terrestre; varredura horizontal e vertical.	Alta densidade de pontos; excelente para ambientes estruturados.	Menor alcance comparado com ALS; sensível a condições atmosféricas.	Navegação robótica, mapeamento de vinhas, monitorização florestal.
LiDAR TOF (Time-of-Flight)	Emite pulsos e mede o tempo de retorno da luz.	Muito preciso; funciona bem em exteriores; não depende da luz solar.	Acurácia reduzida em chuva, nevoeiro ou poeiras.	Navegação autónoma, construção de mapas 3D.
LiDAR de Varredura Contínua (Scanning LiDAR)	Usa espelhos rotativos para gerar um varrimento 2D/3D.	Ótimo para robôs terrestres; alta taxa de atualização.	Pode ter pontos cegos; partes móveis desgastam.	AMRs, robôs agrícolas, controlo de obstáculos.
LiDAR de estado sólido (Solid-State)	Sem partes móveis, usa fases óticas ou microespelhos MEMS.	Muito robusto; pequeno; reduzida manutenção.	Alcance inferior; tecnologia ainda em evolução.	Veículos autónomos, robôs compactos.

2.6.2 Inercial (IMU)

A localização inercial, através de Unidades de Medição Inercial (IMUs) (ver Figura 2-8), é fundamental para estimar a orientação e o movimento do sistema robótico, sobretudo em ambientes onde o sinal de GPS é fraco ou inexistente. Uma

IMU integra tipicamente acelerómetros, giroscópios e, por vezes, magnetómetros, permitindo medir acelerações lineares e velocidades angulares em tempo real e garantindo navegação contínua sem depender exclusivamente de sinais externos.

Contudo, estas medições sofrem de deriva ao longo do tempo, devido ao erro acumulado dos sensores, o que degrada progressivamente a precisão da pose estimada.

Para mitigar esta limitação, a IMU é frequentemente combinada com outras fontes de informação, como GPS, câmaras RGB/estéreo ou sensores LiDAR, recorrendo a técnicas de fusão sensorial baseadas em filtros de Kalman e suas variantes [69-70]. A miniaturização e redução de custo destes dispositivos tem impulsionado a sua adoção em veículos autónomos e robots agrícolas, onde se exige elevada precisão de posicionamento para tarefas como plantação, fertilização e navegação entre culturas [71–72]. Deste modo, a integração de IMUs com outros sensores constitui um elemento-chave para obter uma navegação robusta e fiável em cenários agrícolas desafiantes.

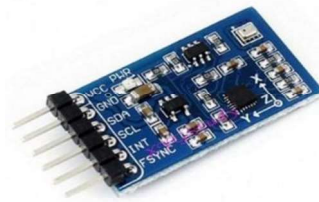


Figura 2-9 IMU [127]

2.6.3 Câmaras

A visão é um dos sentidos mais poderosos do ser humano, devido à quantidade de informação que é absorvida num dado ambiente. Portanto, são compreensíveis 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 *robotic arm manipulation* (ARMs) e com o avanço da inteligência artificial e com a utilização de câmaras

permitem que atualmente estes 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 utilizadas 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 [73];
- Câmaras multiespectral: contém múltiplos sensores e cada um tem 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 [73];
- 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 a 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 [74];
- 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 eletrónico. 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 [74].

Tabela 2-2: Tipos de câmaras.

Tipo de câmara	Princípio de funcionamento	Informação fornecida	Principais vantagens	Principais limitações	Aplicações típicas	Ref.
RGB	Sensor capta luz visível nos canais Vermelho, Verde e Azul, combinando-os por adição de cor.	Imagens 2D em cor (sem informação explícita de profundidade).	Representação semelhante à visão humana; baixo custo; grande disponibilidade.	Não fornece diretamente profundidade; sensível a variações de iluminação e sombras.	Navegação básica, deteção de objetos, visão para IA/ML.	[73]
Multi spectral	Conjunto de sensores com filtros que captam bandas específicas do espectro eletromagnético (visível + não visível).	Imagens em várias bandas espectrais; índices de vegetação e stress hídrico.	Permite extrair informação biofísica das culturas; muito usada em agricultura de precisão.	Custo elevado; maior complexidade de processamento; dados mais pesados.	Monitorização de culturas, deteção de doenças, mapeamento de vigor vegetal.	[73]
Depth	Emissão de luz/laser e medição do tempo de voo (TOF) ou disparidade estereoscópica para estimar distância.	Mapa de profundidade (distância ponto-a-ponto) e, em alguns casos, cor (RGB-D).	Fornecer geometria 3D do ambiente; ideal para navegação, evitamento de obstáculos e SLAM.	Alcance limitado em exteriores muito iluminados (alguns modelos); sensível a superfícies muito brilhantes ou transparentes.	SLAM, planeamento de caminhos, evitamento de obstáculos, mapeamento 3D.	[74]
Térmica	Sensor deteta radiação infravermelha emitida pelos corpos e converte em imagem de intensidade térmica.	Mapa 2D de temperatura relativa (calor emitido pelos objetos).	Independente da iluminação visível; útil em condições de nevoeiro, fumo ou noite; deteta alvos “quentes”.	Menor resolução espacial; custo superior; não fornece profundidade nem cor visível.	Vigilância, inspeção industrial, deteção de pessoas/animais, monitorização de stress térmico.	[74]

2.7 Percepção e mapeamento

2.7.1 ROS

O Robot Operating System 2 (ROS 2) é a evolução do middleware robótico de código aberto ROS 1, amplamente utilizado para o desenvolvimento de aplicações robóticas. Esta atualização visa resolver as limitações do ROS 1, como a falta de suporte para sistemas em tempo real e desafios em sistemas do tipo multi-robôs.

Uma das principais alterações do ROS 2 é a adoção do Data Distribution Service (DDS) como camada principal de comunicação entre processos, melhorando a eficiência e a escalabilidade das aplicações robóticas. Estudos recentes têm explorado o desempenho do ROS 2 em comparação com o ROS 1, especialmente em sistemas multi-robôs.

Por exemplo, a investigação focada na migração de software multi-robô desenvolvido em ROS 1 e ROS 2 revelou que o ROS 2 oferece melhorias significativas em termos de latência e de desempenho geral, devido à sua arquitetura do middleware. Além disso, a composição de nós no ROS 2 permite a integração manual ou dinâmica de nós em processos, mantendo a separação de responsabilidades no código.

Esta abordagem de composição resulta em otimizações de desempenho, especialmente valiosas para sistemas com recursos limitados e pipelines de processamento de sensores, permitindo a execução de tarefas distribuídas que não seriam possíveis no ROS 1 [75].

No contexto de veículos aéreos não tripulados (MAVs), foi proposta uma arquitetura de sensoriamento baseada no ROS 2 para superar desafios de recursos limitados. Utilizando o micro-ROS, uma versão reduzida do ROS 2 projetada para dispositivos com restrições de recursos, os MAVs foram integrados de forma transparente ao mundo ROS 2/DDS, permitindo a comunicação e cooperação com

outros dispositivos habilitados para ROS 2, incluindo estações terrestres e sensores adicionais [76].

A avaliação de diferentes middlewares do ROS 2, como FastRTPS, CycloneDDS e Zenoh, em redes mesh com topologias dinâmicas, destacou o Zenoh como uma solução promissora para aplicações futuras. O Zenoh demonstrou um menor atraso, melhor alcance e uso eficiente de CPU, tornando-se uma opção viável para ambientes extremos, como missões de exploração lunar [77]. Tem-se realizado também pesquisas com foco na análise de tempo e em aperfeiçoamentos baseados em prioridades dos executores multithreaded do ROS 2.

Um framework abrangente de análise de tempo de resposta para processamento em paralelo em executores multithreaded foi proposto, com intuito de garantir um desempenho confiável em tempo real mesmo com complexidade da execução multithreaded em aplicações robóticas críticas [78].

Em suma, o middleware ROS 2 representa um avanço significativo no desenvolvimento de sistemas robóticos, oferecendo melhorias no desempenho, flexibilidade e suporte para aplicações em tempo real. A sua adoção crescente quer na indústria quer nas universidades destaca o seu potencial da robótica. Na Figura 2-10 é possível ver o FrameWork ROS2.

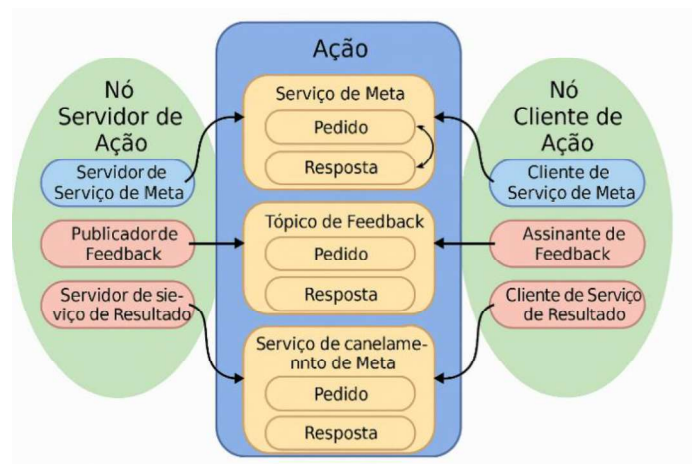


Figura 2-10 FrameWork ROS2 [131]

2.7.2 Odometria Visual

A odometria visual é uma das técnicas fundamentais utilizadas para estimar a posição e a orientação de um sistema robótico a partir de imagens adquiridas por câmaras. Esta técnica tem ganho destaque, especialmente em ambientes onde a utilização de sensores tradicionais, como o GPS ou o IMU, é limitado ou até impossível. A principal vantagem da odometria visual é que permite a visualização contínua da trajetória do sistema robótico utilizando apenas informações visuais, o que permite ser uma opção viável para sistemas autónomos em ambientes não estruturados.

A odometria visual é baseada na análise de sequências de imagens, nas quais as mudanças nas características visuais dos ambientes são utilizadas para estimar o movimento do sistema robótico. Os algoritmos de odometria visual podem ser classificados em dois tipos: monocular e estéreo. No caso da odometria visual monocular, utiliza-se apenas uma câmara para adquirir imagens e realizar estimativa de movimento, enquanto a odometria estéreo faz uso de duas câmaras para calcular a profundidade e melhorar a precisão.

Nos últimos anos, a área de odometria visual teve avanços significativos especialmente a nível da utilização de redes neurais profundas e de sistemas de aprendizagem para melhorar a precisão e a robustez. A utilização destes algoritmos tem permitido melhorar a tolerância a falhas em ambientes complexos, como áreas com pouca textura visual, que anteriormente causavam dificuldades significativas para a odometria visual.

Em 2020, um estudo explorou a integração de redes neurais convulsionais (CNNs) para detetar e rastrear características visuais de alta-fidelidade em tempo real, permitindo uma estimativa de movimento mais robusta e precisa, mesmo em ambientes dinâmicos com iluminação variável [79].

A fusão de odometria visual com outros sensores, como por exemplo os IMUs, também tem sido um tópico de investigação relevante. De acordo com os trabalhos realizados pelo autor [80], a combinação de odometria visual com os IMUs permitiu corrigir erros acumulados e reduzir a deriva visual, aumentando a precisão da navegação em ambientes dinâmicos. A utilização do algoritmo do filtro de Kalman estendido tem sido amplamente utilizado para efetuar a fusão das medições de odometria visual com os dados inerciais, o que resulta num sistema de navegação híbrido mais robusto.

Outro avanço significativo foi a introdução de câmaras com taxa de processamento de imagem em tempo real mais elevadas, que têm permitido a aplicação de odometria visual a velocidades mais elevadas. A capacidade de processar volume elevado de dados visuais em tempo real tem sido crucial para melhorar a precisão da estimativa de posição em ambientes com movimentos ou mudanças do cenário visual [81].

Além disso, a odometria visual monocular continua a ser uma área de grande interesse, porque permite uma solução acessível e compacta. Embora as câmaras estéreo forneçam uma estimativa de profundidade mais precisa, a odometria monocular é eficaz em muitas aplicações, especialmente quando combinada com outras fontes de dados. Por exemplo, uma investigação recente mostrou que ao utilizar as técnicas como a aprendizagem profunda, é possível melhorar significativamente a qualidade da odometria visual monocular, mesmo em cenários de pouca luz ou com pouca textura visual [82].

Portanto, a odometria visual é uma ferramenta poderosa para a navegação, especialmente em cenários onde a precisão e a robustez são essenciais. A integração de técnicas de aprendizagem com a fusão sensorial, continuam a melhorar a eficiência e a confiabilidade, tornando-a possível como uma solução neste tipo de aplicações.

2.7.3 Visual SLAM

A Localização e Mapeamento Simultâneos (SLAM, do inglês Simultaneous Localization and Mapping) continua a ser fundamental na investigação quer da robótica quer da visão computacional, permitindo assim que os sistemas robóticos naveguem em ambientes desconhecidos sem dependência de infraestrutura externa.

Dentro desse contexto, o Visual SLAM (V-SLAM) destaca-se por utilizar apenas câmeras como sensores principais para a estimar a pose, bem como a construção do mapa em que o sistema robótico executa a sua missão [83]. O V-SLAM tem atraído cada vez mais interesse devido à ampla disponibilidade de câmaras e à redução dos custos associados a sensores especializados, tais como o LiDAR.

Os avanços em técnicas de aprendizagem profunda e a otimização de gráficos facilitaram melhorias significativas na precisão e robustez dos algoritmos de SLAM [84]. Os sistemas de V-SLAM geralmente apresentam uma arquitetura composta por módulos distintos, incluindo deteção, rastreamento de características visuais, estimação do movimento, otimização de trajetória e construção do mapa. Os desafios do V-SLAM incluem a robustez a mudanças de iluminação, oclusões e ambientes com baixa textura, onde os métodos tradicionais de correspondência de características podem falhar. Além disso, a escalabilidade para aplicações em escala, como veículos autónomos e a realidade aumentada exige algoritmos eficientes e otimizados [85].

Métodos baseados em formas, como ORB-SLAM, utilizam detetores de pontos-chave para mapear e acompanhar características no ambiente [86]. Por outro lado, abordagens baseadas em aprendizagem de máquina, como DeepVO e DSO (Direct Sparse Odometry), aproveitam redes neurais profundas para realizar inferências diretas sobre a posição e a estrutura do ambiente [87].

O V-SLAM tem sido aplicado em diversas áreas, incluindo navegação, mapeamento 3D e interfaces com a realidade aumentada. O seu desenvolvimento contínuo promete melhorar significativamente a percepção computacional e a autonomia de dispositivos inteligentes no futuro.

2.7.3.1 Orb-SLAM3

O ORB-SLAM3, é uma evolução significativa dos sistemas de Simultaneous Localization and Mapping (SLAM), destacando-se pela sua capacidade de operar em tempo real com diferentes configurações de sensores, incluindo câmaras monoculares, estéreo e RGB-D, além de integrar dados de unidades de medição inercial (IMU) [88]. Esta versatilidade torna o ORB-SLAM3 uma ferramenta promissora para aplicações na agricultura de precisão. No entanto, os sistemas de SLAM revelam dificuldades na parametrização visto serem muito frágeis.

A aparência visual repetitiva dos campos e as mudanças causadas por condições climáticas podem dificultar a detecção em *close Loop*, uma técnica essencial para corrigir desvios acumulados durante a navegação. Para enfrentar essas dificuldades, investigadores propuseram a fusão de dados de sistemas globais de navegação por satélite (GNSS) com medições estéreo e inerciais no pipeline do ORB-SLAM3.

Esta abordagem conhecida como GNSS-estéreo-inercial SLAM, mostrou reduzir erros de pose entre 10% e 30% em comparação com métodos apenas visuais-inerciais ou com acoplamento menos integrado [89].

A aplicação do ORB-SLAM3 não se limita apenas a ambientes agrícolas tradicionais, foram realizados estudos de viabilidade que demonstraram a utilização de drones equipados com câmaras monoculares para inspeção de estufas e instalações pecuárias. Nesses cenários, algoritmos de SLAM visual, como o ORB-SLAM3, foram avaliados quanto à sua precisão e robustez, indicando o seu potencial para operações em ambientes internos complexos [90].

A integração de algoritmos de SLAM visual em robôs agrícolas oferece benefícios significativos, como a capacidade de operar em ambientes não estruturados e a redução da dependência de infraestruturas externas para localização.

No entanto, é fundamental considerar as limitações atuais, como a necessidade de condições de iluminação adequadas e a complexidade computacional envolvida [91].

Pesquisas contínuas visam melhorar a precisão e a eficiência desses sistemas, explorando, por exemplo, técnicas de construção de mapas em tempo real baseadas no ORB-SLAM3.

Em suma, a aplicação do ORB-SLAM3 na agricultura representa um avanço significativo na procura por soluções de navegação autônoma em ambientes complexos. A combinação de dados visuais, inerciais e de GNSS permite superar desafios específicos do setor, contribuindo para o desenvolvimento de sistemas agrícolas mais eficientes e autônomos.

2.7.3.2 OctoMap

O OctoMap é uma das “frameworks” mais utilizadas na robótica para a construção do mapa a três dimensões, com o objetivo de representar simultaneamente os espaços livres, os ocupados e os ainda desconhecidos. Deste modo, é possível criar uma memória facilmente interpretada pelos módulos de navegação e planeamento. É exequível implementar esta “framework” devido a basear-se numa estrutura de dados designada por octree, em que são criados cubos (voxels) que são divididos hierarquicamente de modo a diferenciar a zona.

O OctoMap é uma das abordagens que permite recriar ambientes externos extensos com resolução variável, com o objetivo de evitar manter uma grelha a três dimensões densa e com resolução fixa em toda a cena [92].

Do ponto de vista do utilizador, o OctoMap mantém a ideia clássica de criar uma grelha de ocupação 2D para três dimensões. Cada voxel guarda a probabilidade de estar ocupado, $p(mi)$, que é calculada a partir dos sensores (por exemplo, câmara RGB-D, LiDAR ou sonar). No lugar de apenas definir dois estados binários (ocupado/livre), este mantém a probabilidade de estar ocupado e atualiza a posição de ocupação de forma sequencial conforme a chegada de novos dados adquiridos pelos sensores. Assim é possível combinar informação acerca dos obstáculos e sobre a área livre para navegação, permitindo a correção das leituras com ruído ao longo do tempo e as incertezas nas zonas mais observadas [93].

Se a zona analisada for uniforme (por exemplo, claramente livre ou ocupada), a cada área criada é possível ser mantida uma resolução mais reduzida. No entanto, se a região tiver uma geometria complexa, a subdivisão será contínua, mas a resolução irá aumentar até ao ponto que foi definida pelo utilizador. Esta exposição multi-resolução permite interpretar diferentes tipos de áreas, como por exemplo, áreas de colisão muito próximas ao sistema robótico ou ao planeamento da rota (ou missão?) para zonas mais afastadas, usando o mesmo mapa ajustando dinamicamente o nível de detalhe da leitura [94].

A organização dos octantes contribui para a eficiência em memória e para a taxa de execução. Em ambientes externos extensos, como os campos agrícolas ou áreas florestais, iremos ter a maior parte das zonas vazias, o octree evita a colocação de voxels em regiões pouco relevantes, condensando recursos nas áreas que exigem maior resolução como obstáculos ou estruturas de interesse.

Experiências recentes têm explorado o desenvolvimento desta ideia de forma a criar mapas mais robustos de elevada resolução, com aplicação na agricultura de precisão, e na navegação de veículos UAVs [95].

A modelação do estado de cada voxel “mi” como uma variável aleatória binária (ocupado ou livre) e é aplicado um filtro de Bayes binário para atualizar as sucessivas probabilidades de ocupação com base nas leituras consecutivas dos

diferentes sensores ($Z_{1:t}$). A formulação matemática associada a este filtro é dada por:

$$P_t(mi|z_{1:t}) = \frac{p(Z_t|mi) * P_{t-1}(mi|Z_{1:t-1})}{p(Z_t|Z_{1:t-1})} \quad (20)$$

Que resulta da relação entre a probabilidade anterior e a veracidade da nova medição. Na realidade o OctoMap recorre a uma representação em log-odds para otimizar as atualizações ao mapa sendo numericamente confiável. O log-odds da ocupação é descrito da seguinte forma:

$$lt(mi) = \log \frac{pt(mi)}{1 - pt(mi)} \quad (21)$$

A atualização constante de forma incremental, em que o novo valor do log-odds é obtido somando-se com a medição anteriormente acumulada. Após cada atualização do log-odds, a probabilidade de ocupação é possível obter a partir da fórmula:

$$pt(mi) = \frac{1}{1 + \exp(-lt(mi))} \quad (22)$$

Além disso, os limites de saturação designados de clamping são aplicados aos valores de log-odds, assim impossibilita que a probabilidade seja 0 ou 1. É necessário manter uma ínfima incerteza possibilitando retificações futuras, verificando-se que observações consistentes das novas alterações contraditórias continuam a ter a capacidade de alterar os estados dos voxels de forma progressiva.

A atualização do mapa é realizada com base na integração dos dados dos diferentes sensores. Cada avaliação é compreendida como um conjunto de feixes

emitidos pelos sensores desde a sua origem até aos pontos de observação na nuvem de pontos ou na imagem de profundidade.

Quaisquer voxels atravessados por algum tipo de sensor até ao ponto de nova colisão são atualizados como espaço livre, fortalecendo a possibilidade de ser uma zona livre, enquanto o voxel que foi atravessado será indicado como ocupado. Com a informação adquirida é possível distinguir entre uma zona livre ou ainda não conhecida, adquirindo informação positiva ou negativa, obtida a partir das medições [96].

No ponto de vista computacional, com a utilização de sensores com elevada resolução, para a identificação dos voxels pode tornar-se uma operação intensiva.

Assim, os trabalhos recentes têm explorado a forma de aceleração significativa deste tipo de processo, recorrendo a melhores placas gráficas e a técnicas de caching voxel para evitar atualização redundantes.

É possível conservar a filosofia probabilista do OctoMap introduzindo melhoria importante na eficiência do tempo de execução em cenários de escala elevada [97]. O OctoMap diferencia-se dos diferentes algoritmos de mapeamento devido à representação probabilista, à sua eficiência e à facilidade da consulta deste.

O mapa contruído pelo OctoMap permite responder facilmente às questões de operação típicas em sistemas robóticos como a navegação autónoma destes, zonas de colisão, como também na criação de mapas de custo 2D com projeção do volume identificado pelo sensor ou ainda na verificação das zonas ainda não exploradas para o planeamento da procura.

Em aplicações modernas, o OctoMap é frequentemente utilizado como representação 3D sobre a qual são construídos path-planners, algoritmos de exploração e sistemas multi-robô que interligam os mapas individuais num mapa global partilhado [98].

As limitações do OctoMap destacam-se no custo de atualização em cenários com taxas de amostragem muito elevadas e na dificuldade em lidar com ambientes altamente dinâmicos, onde objetos em movimento introduzem inconsistências no mapa. Outra limitação do OctoMap é o facto de representar apenas a ocupação e não adquirir explicitamente superfícies contínuas ou os detalhes geométricos, o que levou ao desenvolvimento de abordagens híbridas e de novas estruturas de dados para aplicações que exigem modelos geométricos com mais informação [99].

Apesar das limitações, o OctoMap continua a ser uma das ferramentas de referência para mapeamento 3D em robótica, pela robustez do modelo probabilístico, pela capacidade de representar de forma consistente espaço livre, ocupado e desconhecido, e pela integração consolidada no ecossistema ROS [100].

No contexto deste Trabalho de Projeto, o OctoMap é utilizado como base para a representação volumétrica do ambiente agrícola, integrando as medições de profundidade fornecidas pela câmara RGB-D e a estimativa de pose do robô, e o mapa 3D compatível com os módulos de navegação e planeamento.

3 SIMULADOR DO SISTEMA ROBÓTICO

A simulação faz parte de uma etapa fundamental no desenvolvimento e validação de sistemas robóticos, possibilitando assim, testar e otimizar o desempenho do protótipo em ambiente virtual antes da sua implementação física.

Esta abordagem reduz significativamente os custos e riscos associados à fase experimental, enquanto possibilita uma análise detalhada do comportamento dinâmico, da interação com o meio envolvente e da resposta dos algoritmos de controle.

O ambiente de simulação desenvolvido baseia-se na integração do Gazebo e do ROS2 (Robot Operating System), que ofereceu uma plataforma robusta e flexível para a modelação, controlo e monitorização do sistema robótico. O modelo tridimensional do sistema robótico, foi convertido para o formato compatível com o simulador e equipado com sensores virtuais que replicam o comportamento real de dispositivos como a câmara RGB-D, o IMU.

Através desta configuração, foi possível implementar e testar algoritmos de navegação autónoma, localização e controlo de movimento, verificando a sua estabilidade e eficiência em diferentes cenários operacionais.

A simulação permite ainda a avaliação e sintonização dos controladores PID descritos no capítulo anterior, comparar o seu desempenho face a variações de terreno e na presença de obstáculos.

3.1 Desenho 3D

O modelo tridimensional do sistema robótico foi desenvolvido integralmente de modo a representar as estruturas físicas e os parâmetros dinâmicos de um sistema robótico. Deste modo, implementou-se a definição dos ficheiros URDF (Unified Robot Description Format) e SDF (Simulation Description Format), formatos

amplamente utilizados no âmbito da robótica conforme se mostra nas Figuras 3-1 e 3-2.

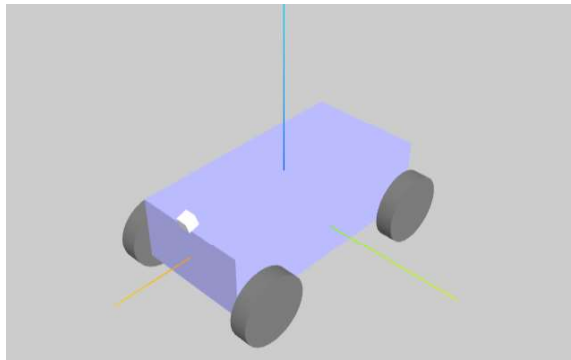


Figura 3-1 Desenho 3D do sistema robótico

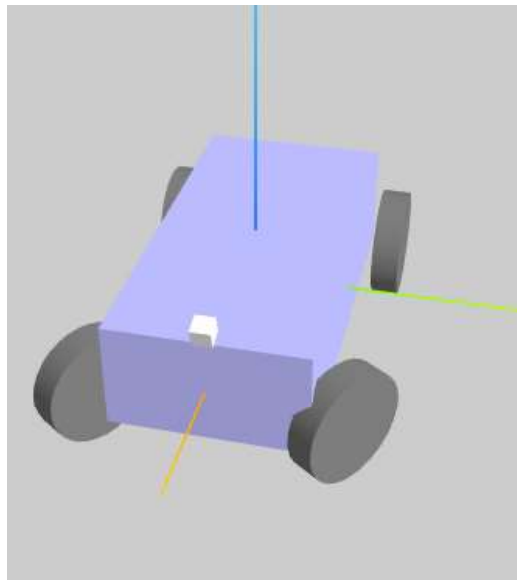


Figura 3-2 Desenho 3D do sistema robótico com um ângulo de -60°

O ficheiro URDF permite descrever a configuração cinemática do sistema robótico, incluindo os seus links (corpos rígidos), joints (articulações), motores e a respetiva hierarquia estrutural. Cada componente foi definida com recurso às propriedades físicas específicas, como a massa, a inércia, as dimensões e os materiais, permitindo assegurar um comportamento realista durante a simulação. Para a modelação visual, recorreu-se a formatos geométricos primitivos tipo (caixas,

cilindros e esferas), que foram combinados de modo a representar o corpo principal do robô e os seus subsistemas mecânicos, conforme se mostra na Figura 3-3.

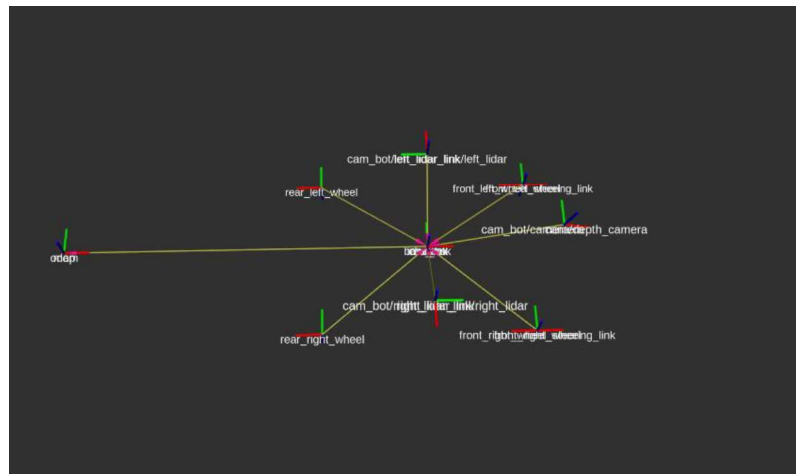


Figura 3-3 Corpos rígidos do sistema robótico

O ficheiro SDF, por sua vez, descreve o ambiente de simulação e complementa o modelo do sistema robótico com parâmetros adicionais, como texturas, sensores e propriedades do terreno. Permite ainda, configurar as propriedades físicas do solo, incluindo fricção, rugosidade e densidade, de forma a reproduzir as condições típicas de um terreno agrícola. Como o objetivo é avaliar o desempenho do robô em cenários realistas, especialmente no que diz respeito à tração, estabilidade e controlo de trajetória, este nível de detalhe é essencial para uma boa avaliação, conforme se mostra na Figura 3-4.

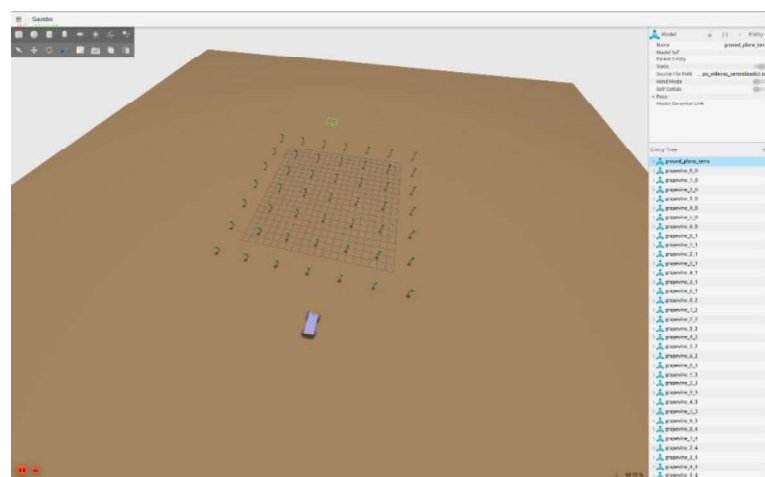


Figura 3-4 Terreno desenvolvido para a simulação

O modelo desenvolvido integra também sensores virtuais que permitem a recolha de dados de perceção durante a simulação, nomeadamente:

- Sensor inercial (IMU), que fornece informação da aceleração e orientação do robô;
- Câmara RGB ou RGB-D, utilizada para a visualização e algoritmos de navegação, conforme ilustrado na Figura 3-5.

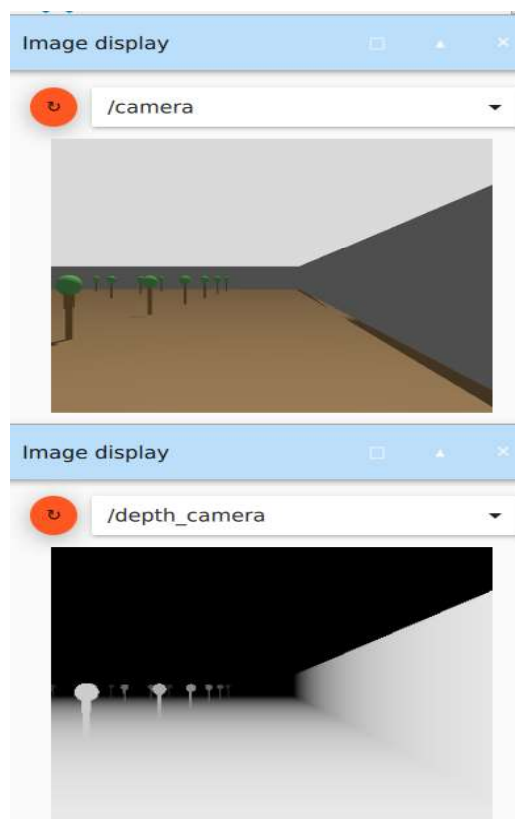


Figura 3-5 Imagem da câmara RGB-D

Para assegurar a comunicação entre o ambiente de simulação e os algoritmos de controlo e navegação é imprescindível a integração destes sensores no modelo URDF/SDF, pois permite a publicação direta dos respetivos tópicos no ROS2.

Assim, o sistema robótico simulado pode operar em tempo real sob as mesmas condições de um sistema físico, o que valida a arquitetura de controlo e o desempenho dos algoritmos antes da implementação prática.

Desta forma, a modelação tridimensional baseada em URDF e SDF constitui uma etapa fundamental para a simulação e validação do sistema robótico, garantindo uma representação fiel das suas características estruturais e comportamentais, bem como uma integração completa com o ecossistema ROS2-Gazebo.

3.2 Ambiente de Simulação

Para desenvolver o ambiente de simulação utilizou-se a plataforma Gazebo, uma ferramenta de simulação física amplamente utilizada na área da robótica. Esta permite recriar com elevado realismo as interações entre sistemas robóticos e o meio envolvente, bem como, avaliar o comportamento dinâmico, quanto à estabilidade e à resposta do sistema robótico em diferentes condições operacionais, antes da construção do protótipo físico. O Gazebo possui um motor físico avançado que simula fenómenos como gravidade, colisões, atrito e forças de contacto, apresentando uma representação fiel das condições reais de funcionamento. Esta característica é vital para o estudo de veículos agrícolas autónomos devido ao impacto que os comportamentos deles têm nos efeitos do solo, na irregularidade do terreno e na distribuição de peso.

Para a criação do cenário de simulação, definiu-se um ambiente agrícola virtual, que representa uma superfície de terreno com variações topográficas e diferentes tipos de solo. De forma a reproduzir as condições típicas de um campo de cultivo, configurou-se a textura e as propriedades físicas do solo, integrando aspetos como a rugosidade, atrito e resistência mecânica. Estes foram ajustados aos parâmetros de mobilidade do sistema robótico. Para testar a capacidade de navegação e a robustez do sistema em situações complexas, o ambiente inclui ainda elementos visuais e físicos como zonas de vegetação, obstáculos e delimitações de área, conforme se mostra na Figura 3-6.

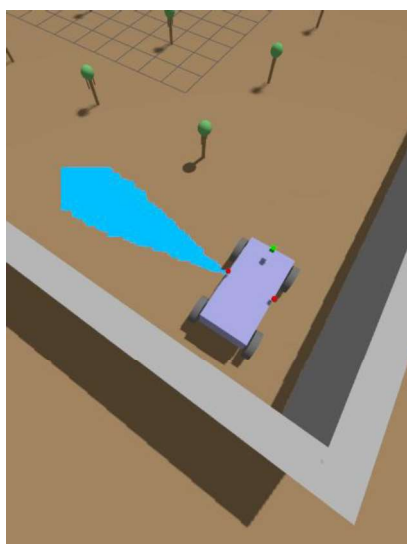


Figura 3-6 Imagem do sistema robótico e do ambiente virtual

A simulação completa da locomoção e interação do sistema robótico com o terreno foi obtida através da implementação do sistema robótico no ambiente como um modelo dinâmico compatível com o motor físico do Gazebo. Para garantir um comportamento coerente com o esperado em operação real, definiram-se os parâmetros de massa, inércia, coeficientes de atrito e limites de esforço das juntas, juntamente com os plugins físicos responsáveis por controlar a movimentação, gerir a dinâmica das rodas e permitir a atuação sobre os motores simulados.

O ambiente inclui ainda fontes de iluminação e texturas ambientais que reproduzem as condições de luz natural, permitindo a utilização de sensores virtuais de visão com resultados comparáveis aos obtidos em campo. Estas condições são particularmente relevantes para algoritmos de percepção e navegação visual, que dependem dos contrastes e da luminosidade para detetar obstáculos e planejar trajetórias.

A flexibilidade da plataforma Gazebo possibilitou a realização de diferentes cenários experimentais, através da variação de parâmetros como a inclinação do terreno, a densidade de obstáculos e a fricção do solo. A diversidade de condições permitiu ainda testar a estabilidade, a resposta dos controladores e a capacidade de adaptação do sistema robótico a diferentes contextos agrícolas.

Assim, o ambiente de simulação desenvolvido no Gazebo constitui uma ferramenta essencial para a validação virtual do sistema robótico agrícola, pois assegura que os algoritmos de controle e as decisões de projeto sejam avaliados em condições controladas, realistas e reproduzíveis.

3.3 Arquitetura de comunicação

A simulação foi implementada em Gazebo com integração em ROS 2, permitindo executar a simulação do robô virtual com sensores e atuar sobre o sistema através de tópicos. O modelo do robô (URDF/SDF) define a cinemática/dinâmica e os sensores (IMU e câmara RGB-D), enquanto o ambiente simulado representa o cenário agrícola e as propriedades físicas do terreno.

Durante a execução, o simulador disponibiliza odometria e transformações (TF), e a percepção (nuvem de pontos) é usada para suportar a construção de mapa e a navegação. A Tabela 3-1 resume a parametrização relevante para reprodutibilidade (software, parâmetros de física, taxas de sensores e tópicos principais).

Tabela 3-1: Parametrização do simulador.

Componente	Parâmetro	Valor
Software	ROS 2	Humble
Software	Gazebo	Gazebo Sim Fortress
Simulação (Física)	<code>`max_step_size`</code>	0.01 s
Simulação (Física)	<code>`real_time_update_rate`</code>	100 Hz
Simulação (Física)	<code>`real_time_factor`</code>	~1.0
Modelo do sistema robótico	Tipo de direção	Ackermann
Terreno	Fricção (μ)	0.9
Sensores	IMU — tópico	<code>`/imu`</code>
Sensores	IMU — frequência	100 Hz
Sensores	RGB — tópico	<code>`/camera`</code>
Sensores	Depth — tópico	<code>`/depth_camera`</code>
Sensores	PointCloud — tópico	<code>`/depth_camera/points`</code>
Sensores	RGB-D — frequência	30 Hz
Frames (TF)	Frame base	<code>'chassis'</code>
Frames (TF)	Odom e mapa	<code>`odom`, `map`</code>
Odometria	Fonte	Plugin do simulador
Odometria	Tópico	<code>`/odom`</code>
Odometria	Frequência	100 Hz
Navegação	Planeador	SmacPlanner2D / outro
Navegação	Controlador	Vector Pursuit / outro
Mapeamento	Resolução do mapa	0.9
Mapeamento	Alcance	10 m

4 RESULTADOS DO SISTEMA ROBÓTICO

4.1 Objetivos dos ensaios

O conjunto de ensaios apresentados neste capítulo têm por finalidade validar a eficiência da arquitetura de navegação autónoma e de controlo desenvolvido. Permitem ainda analisar a fiabilidade do mapa tridimensional fornecido pelo OctoMap, onde os parâmetros foram testados em cenário representativo equivalente à operação agrícola. Quanto ao mapeamento pretende-se classificar o mapa 3D que é construído indicando se é ou não adequado para as tarefas de navegação e planeamento.

Os resultados apresentados ao longo deste capítulo ocorreram em ambiente de simulação utilizando o modelo cinemático do sistema robótico descrito anteriormente, tendo por objetivo representar o sistema robótico real.

Após a análise deste capítulo será possível retirar conclusões acerca da implementação física futura, exigindo sobretudo a substituição das fontes de dados ideais do simulador pelos sensores reais com ruído e incertezas.

4.2 Definição da experiência

As simulações foram realizadas em ambiente de simulação no Gazebo, com a interação do ROS 2 como camada de integração e com NAV2 como núcleo de planeamento de rota e controlo de movimento. O sistema robótico foi idealizado como um veículo agrícola com cinemática do tipo Ackermann, utilizando o mesmo conjunto de *links* e *joints* definidos para um sistema robótico real, permitindo assim garantir a consistência entre os dois contextos.

Nas diferentes simulações realizadas, o sistema robótico idealizado é inicializado numa pose conhecida e recebe um objetivo final sobre uma combinação de pontos espaciais a percorrer (*waypoints*). A odometria que é utilizada para todos os

algoritmos utilizados vêm diretamente do simulador, com origem na integração das equações cinemáticas. Sendo a odometria publicada num tópico que é utilizada como uma variável fulcral para determinar a estimação de posição do sistema robótico. De igual forma a câmara RGB-D virtual fornece imagens RGD, de Depth e ainda a *pointcloud*, esta irá alimentar os tópicos do OctoMap permitindo assim a ativação do módulo de mapeamento.

A parametrização do NAV2 foi construída de forma a ser constante ao longo dos diferentes ensaios, variando as trajetórias de referência e as diferentes posições que o sistema robótico possa ter. Na Figura 4.1 está representado o cenário de simulação das simulações realizadas em Gazebo, onde é possível observar a posição inicial do sistema robótico, as linhas de cultivo e os obstáculos estáticos ainda considerados.

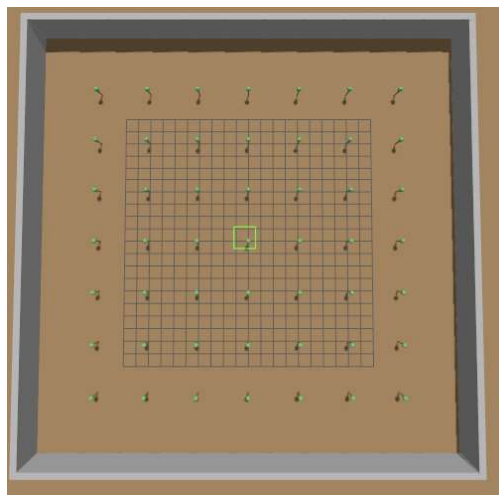


Figura 4-1 Ambiente de simulação

Os parâmetros relevantes do sistema de navegação, desde a velocidade nominal de operação, as frequências de atualização do controlador e ainda as principais opções de configuração do NAV2, estão resumidos na Tabela 4-1.

Tabela 4-1: Configuração NAV2.

Categoria	Parâmetro / plugin	Símbolo / nome nav2	Valor	Unidade / observações
Cinemática	Velocidade linear nominal	Vnom	0,5	$M \cdot s^{-1}$ (valor de referência comandado; velocidades médias $\approx 0,47 m \cdot s^{-1}$ nos ensaios)
Cinemática	Velocidade linear máxima	Max_vel_x	0,6	$M \cdot s^{-1}$ (limite imposto ao controlador)
Cinemática	Velocidade linear mínima	Min_vel_x	0	$M \cdot s^{-1}$
Cinemática	Velocidade angular máxima	Max_vel_theta	0,6	$Rad \cdot s^{-1}$
Cinemática	Aceleração linear máxima	Acc_lim_x	0,5	$M \cdot s^{-2}$
Cinemática	Aceleração angular máxima	Acc_lim_theta	0,8	$Rad \cdot s^{-2}$
Controlador	Frequência de atualização do controlador	Controller_frequency	20	Hz (taxa de cálculo da velocidade de comando)
Planeador	Frequência de replaneamento global	Planner_frequency	1	Hz
Costmaps	Frequência de atualização da costmap local	Update_frequency (local)	10	Hz
Costmaps	Frequência de publicação da costmap local	Publish_frequency (local)	10	Hz
Costmaps	Frequência de atualização da costmap global	Update_frequency (global)	5	Hz
Costmaps	Resolução da grelha de ocupação	Resolution	0,05	M/célula
Costmaps	Tamanho da costmap local	Width, height (local)	10×10	M (janela centrada no robô)
Costmaps	Tamanho da costmap global	Width, height (global)	34×34	M (abrange toda a vinha de simulações)
Controlador	Tipo de controlador	Vector pursuit controller	–	Controlador cinemático para ackermann
Controlador	Raio mínimo de curvatura	Min_turning_radius	1,5	M (coerente com a geometria do cam_bot)
Planeador	Planeador global	Nav2_smac_planner/smacplanner2d	–	Planeador a* em 2d
Planeador	Número máx. De iterações a*	Max_iterations	100000	–
Bt / navegação	Behaviour tree de navegação	Navigate_w_replanning_and_recovery.xml	–	Bt padrão do nav2 com replaneamento e recoveries
Bt / navegação	Tolerância de posição ao objetivo	Xy_goal_tolerance	0,1	M
Bt / navegação	Tolerância de orientação ao objetivo	Yaw_goal_tolerance	5	Graus
Frames	Referencial global	Global_frame	Map	–
Frames	Base do robô	Robot_base_frame	Base_link	–
Frames	Referencial de odometria	Odom_frame	Odom	–
Transformações	Tolerância para tf	Transform_tolerance	0,2	S

4.3 Cenários de teste

Para analisar o comportamento do sistema nas diferentes situações, foram definidas diferentes trajetórias de modo a analisar o comportamento em cada uma das situações, tentando ter condições típicas da operação em terrenos agrícola. Nas diferentes simulações, o sistema robótico tem por objetivo percorrer as linhas de cultivo já pré-definidas, realizando inversão de marcha e contorno de obstáculos sempre que necessário.

Na primeira simulação pretende-se observar o funcionamento do controlador e do OctoMap num deslocamento através de um conjunto de waypoints numa trajetória retilínea. Assim, é possível avaliar o funcionamento da trajetória sem perturbações. A Figura 4-2 apresenta a disposição geométrica da simulação1, cenário e a trajetória pretendida.

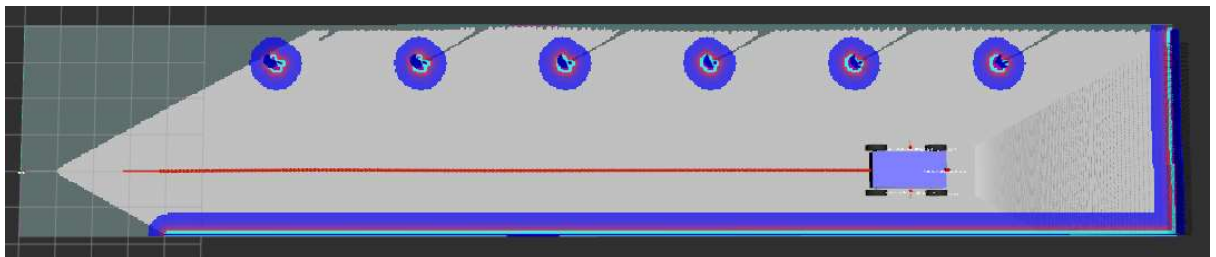


Figura 4-2 Trajetória retilínea

No segundo teste, o sistema robótico tem o objeto de realizar o contorno do campo de simulação, descrevendo um trajeto fechado que delimita a área de operação. Através deste ensaio pretende-se apreciar a capacidade de resposta do NAV2 em coordenadas com curvas sucessivas, assegurando assim o funcionamento contínuo da trajetória multiangular sob as restrições cinemáticas do modelo Ackermann.

Ao longo do percurso, o OctoMap é atualizado a partir das sucessivas medições da câmara RGB-D, assim é possível observar a formação progressiva das

fronteiras do ambiente, a coerência do mapa em três dimensões nas áreas de transições entre os segmentos retilíneos e curvilíneos.

O sistema robótico deve percorrer o limite do cenário, iniciando e terminando no mesmo ponto, sem a necessidade de novo planeamento. Na Figura 4-3 é possível visualizar a disposição do cenário e a trajetória realizada com a referência inicial correspondente, demonstrando os quatro movimentos de 90° assim como os segmentos laterais que compõem este percurso.

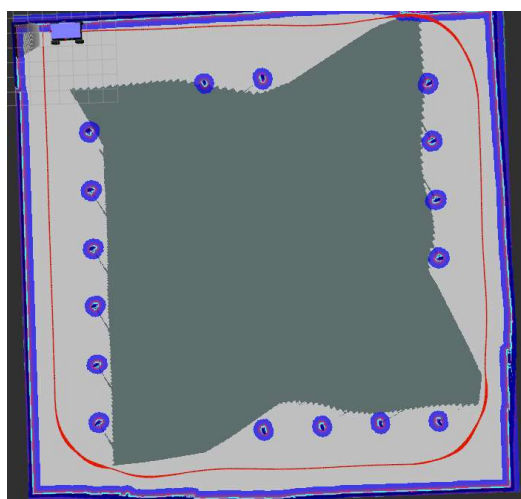


Figura 4-3 Trajetória em torno do cenário

No terceiro teste, o sistema robótico percorre todas as fileiras de videiras do campo de simulação, progredindo ao longo de cada corredor e realizando as respectivas transições entre as linhas de cultivo. Esta simulação permite avaliar a capacidade da navegação e manter um seguimento estável dentro de corredores estreitos, assegurando simultaneamente a execução correta das manobras de mudança de fileira, que envolvem as curvaturas mais acentuadas e exigem uma maior precisão sob as restrições cinemáticas do modelo Ackermann.

À medida que o sistema robótico progride em cada fileira, o OctoMap é atualizado com base na profundidade adquirida através da câmara RGB-D, permitindo assim observar uma formação coerente das bandas de voxels correspondentes às videiras. As transições entre fileiras revelam-se particularmente

relevantes, porque colocam em evidência a interação entre o planejador global responsável pela mudança de linha e o planejador local que ajusta continuamente a trajetória em função do espaço navegável disponível.

Ao completar este objetivo, o sistema robótico percorre toda a extensão das fileiras, regressando ao ponto de partida após cobrir a totalidade da área cultivada.

A Figura 4-4 apresenta a geometria das fileiras e a trajetória descrita pelo sistema robótico, evidenciando tanto o alinhamento longitudinal mantido ao longo dos corredores como as curvas de inversão realizadas nas extremidades de cada linha.

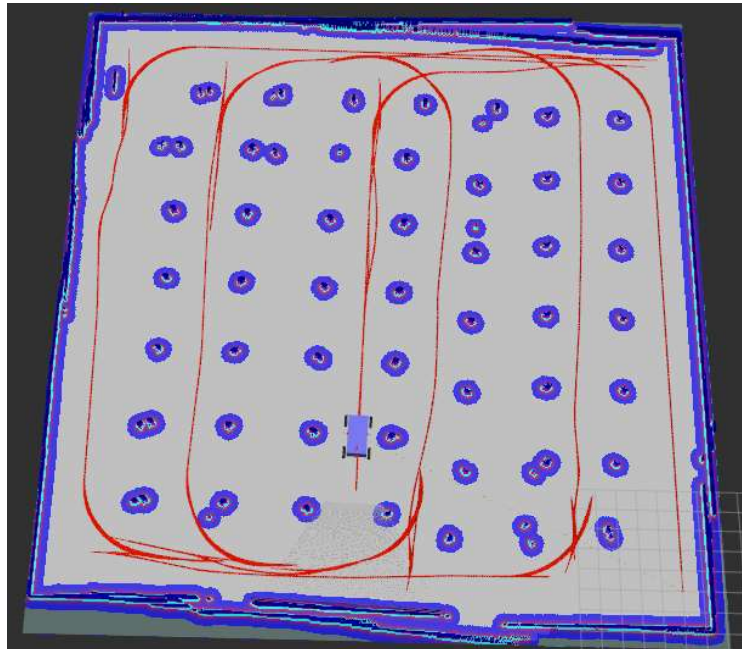


Figura 4-4 Trajetória no cenário

4.4 Métricas de avaliação

Para medir o desempenho do sistema, foram definidas duas métricas, uma para a navegação (seguimento da trajetória) e a outra para o mapeamento (qualidade do mapa criado).

Para o seguimento da trajetória, foi considerado o erro da posição entre a posição atual do sistema robótico e a posição de referência ao longo do tempo. É possível decompor este erro numa componente lateral, medindo a distância

perpendicular entre o sistema robótico e a trajetória definida e ao longo da sua trajetória.

Para realizar esta análise foi utilizada uma métrica de erro quadrático médio (RMSE) da posição no plano, calculada sobre o percurso como também o erro máximo registado.

Além da informação apresentada, são classificados os tempos de missão (intervalo de tempo entre o movimento inicial e o tempo de chegada), de modo a obter a percentagem de waypoints alcançados dentro de uma tolerância pré-definida, e o número de replaneamentos de rota efetuados pelo algoritmo de navegação ao longo de cada teste.

A análise tem em consideração a densidade e consistência do mapa 3D criado pelo OctoMap. As métricas consideradas no mapeamento são a fração do volume do cenário visualizado pelo sistema robótico.

No Vector Pursuit Controller, o parâmetro k representa a constante que relaciona a dinâmica rotacional com a translacional do robô, influenciando o compromisso entre correção de orientação e progressão ao longo da trajetória, valores mais elevados favorecem a translação, enquanto valores mais baixos favorecem a rotação.

4.5 Resultados de navegação

4.5.1 Seguimento de trajetória no cenário de referência

Na simulação n.º 1, o sistema robótico é preparado para realizar uma trajetória retilínea, sem obstáculos, iniciando a navegação numa extremidade do campo e terminando-a após completar todas as passagens previstas até chegar ao final do mapa, conforme se pode observar na Figura 4-5.

É possível verificar que a trajetória realizada é muito semelhante à trajetória ideal, à qual não representa desvios significativos, e que a cinemática Ackermann não

apresenta restrições para este tipo de movimentos, os erros de posição estão representados na tabela 4-2.

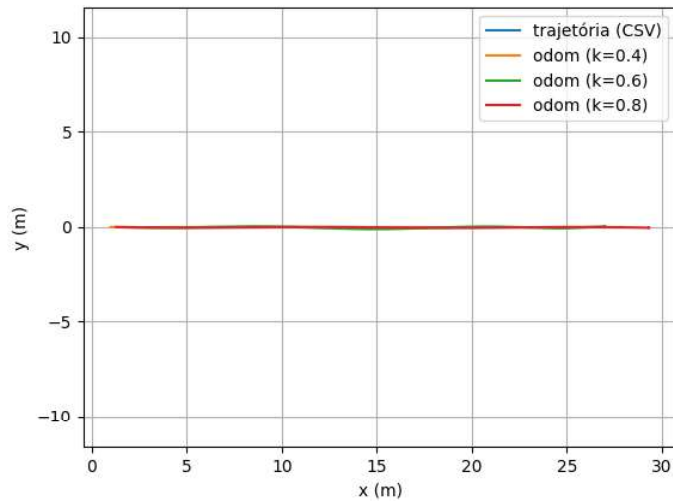


Figura 4-5 Trajetória retilínea

Tabela 4-2: Dados simulação n.º 1.

Métrica	Teste 1 k=0,4	Teste 2 K=0,6	Teste3 K=0,8
Número de amostras	2018	1380	1089
Tempo total de missão (s)	20.170	13.790	10.880
Distância total percorrida (m)	27.090	25.320	25.067
Velocidade média (m/s)	1.343	1.836	2.580
RMSE da posição (m)	0.0867	0.0563	0.0220
Erro máximo (m)	0.166	0.121	0.0432
Waypoints atingidos (%)	100	100	100
Número de replaneamentos	0	0	0

4.5.2 Navegação em torno da plantação

A Figura 4-6 mostra a trajetória do sistema robótico, designada simulação n.º 2, onde se observa que, na generalidade do percurso, a trajetória está aproximada de forma satisfatória da trajetória desejada, com desvios mais significativos na zona

das curvas, onde a curvatura é mais elevada, constatando-se que a cinemática Ackermann impõe restrições ao movimento do sistema robótico.

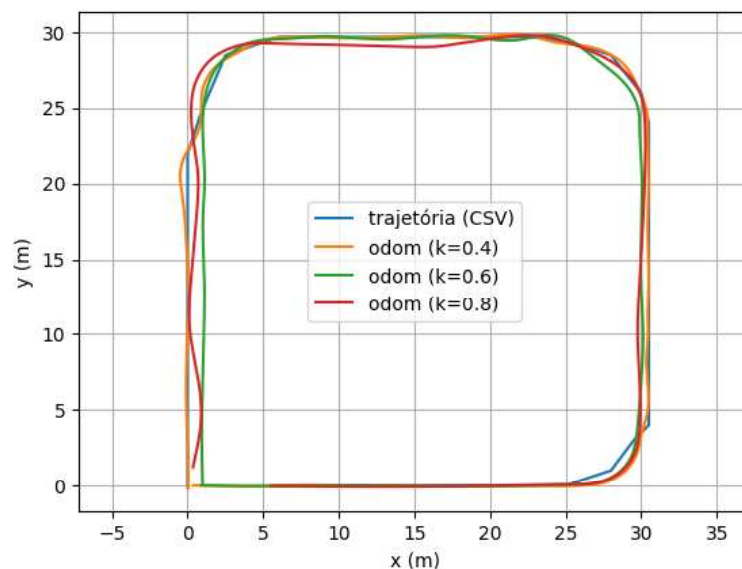


Figura 4-6 Trajetória em torno do cenário

O erro de posição médio e máximo, o tempo total de missão e a percentagem de waypoints atingidos, encontram-se resumidos na Tabela 4-3.

Tabela 4-3: Dados simulação n.º 2.

Métrica	Teste 1 k=0,4	Teste 2 K=0,6	Teste3 K=0,8
Número de amostras	10253	7744	6992
Tempo total de missão (s)	102.520	77.43	69.910
Distância total percorrida (m)	112.874	110.705	106.352
Velocidade média (m/s)	1.101	1.430	1.521
RMSE da posição (m)	0.401	0.618	0.499
Erro máximo (m)	2.996	2.9980	1.117
Waypoints atingidos (%)	100	100	100
Número de replaneamentos	0	0	0

De forma geral, o erro médio obtido é um valor baixo, sugerindo que a combinação entre o controlador de seguimento e o NAV2 é adequada em condições próximas do ideal. Os resultados demonstram ainda que, em cenários muito simples,

não se verificam replaneamentos significativos, o que é coerente com a ausência de obstáculos.

4.5.3 Navegação em toda a plantação

No simulação n.º 3, o sistema robótico é instruído a percorrer todas as fileiras de videiras presentes no ambiente de simulação, deslocando-se longitudinalmente ao longo de cada corredor e realizando as respetivas mudanças de linha nas extremidades, conforme se mostra na Figura 4-7 evidenciando a alternância entre as diferentes fileiras e o padrão de movimento característico deste tipo de trajetória em campos agrícolas.

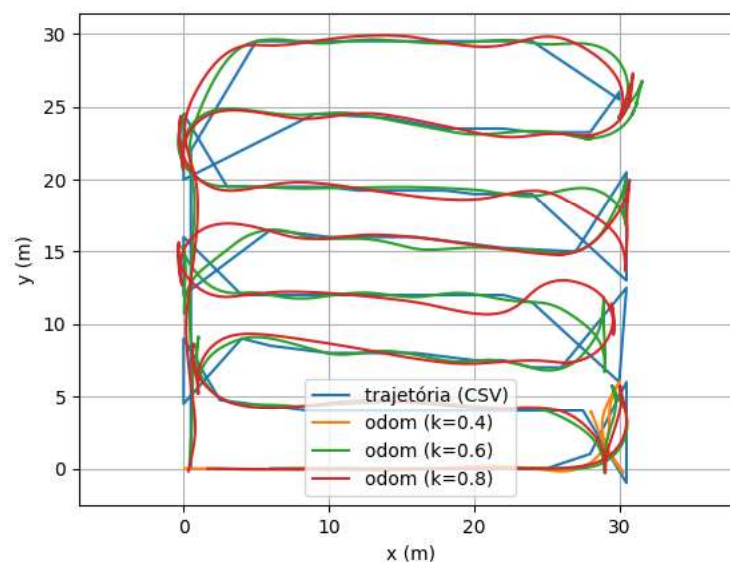


Figura 4-7 Trajetória de toda a vinha

Observa-se que, ao longo de cada fileira, o sistema robótico se mantém tendencialmente centrado no corredor, apresentando apenas pequenas oscilações laterais associadas à proximidade das videiras e à discretização do mapa de ocupação.

Estes desvios mais pronunciados ocorrem nas transições entre fileiras, onde a execução de curvas mais apertadas exige maior esforço de correção por parte da cinemática Ackermann e do planeador local do NAV2.

O gráfico da evolução do erro lateral ao longo do tempo pode realçar o incremento do erro nos instantes que correspondem às manobras de inversão, bem como as manobras de alinhamento com a fileira seguinte, seguidos de uma rápida estabilização quando o sistema robótico retoma o percurso retilíneo. Este comportamento confirma a capacidade do controlador em recuperar o alinhamento após as curvas apertadas e manter o seguimento preciso em corredores estreitos.

Os erros de posição, o tempo total de missão, a percentagem de waypoints alcançados e o número de replaneamentos efetuados durante o percurso, encontram-se sintetizados na Tabela 4-4.

Tabela 4-4: Dados simulação n.º 3.

Métrica	Teste 1 k=0,4	Teste 2 K=0,6	Teste3 K=0,8
Número de amostras	48450	41506	39740
Tempo total de missão (s)	484.490	415.050	397.740
Distância total percorrida (m)	357.911	356.821	360.310
Velocidade média (m/s)	0.739	0.860	0.907
RMSE da posição (m)	0.5616	0.612	0.623
Erro máximo (m)	2.113	2.743	2.7409
Waypoints atingidos (%)	100	100	100
Número de replaneamentos	0	0	0

De forma geral, o erro lateral mantém-se dentro de limites aceitáveis para navegação em vinha, onde a operação ocorre em espaços reduzidos e exige correções contínuas de trajetória. A elevada percentagem de waypoints atingidos e a conclusão integral da missão demonstram que o sistema é capaz de percorrer sequencialmente todas as fileiras, contornando as irregularidades do ambiente simulado sem comprometer a segurança ou a estabilidade do movimento. Estes resultados reforçam a adequação da combinação entre o NAV2, o modelo cinemático Ackermann e o mapa tridimensional criado pelo OctoMap para operações autónomas em ambiente agrícola estruturado.

4.6 Resultados do mapeamento 3D com OctoMap

Nesta secção, são analisados os mapas tridimensionais criados pelo OctoMap ao longo das três simulações realizadas, com o objetivo de avaliar a coerência da representação volumétrica do ambiente e a sua utilidade para suportar a navegação autónoma. A análise incide sobretudo na consistência entre espaço livre, ocupado e desconhecido, na fidelidade geométrica das fileiras simuladas e na estabilidade do mapa durante percursos prolongados. A densidade de voxels ocupados é muito reduzida, refletindo a inexistência de vegetação ou estruturas próximas.

A Tabela 4-5 apresenta os indicadores globais do mapa, incluindo a fração de volume observado e o número total de voxels marcados como ocupados e livres.

Tabela 4-5: Dados visuais do OctoMap.

Indicador	Simulação n.º 1 – trajetória retilínea	Simulação n.º 2 – contorno do cenário	Simulação n.º 3 – fileiras da vinha
Grau de cobertura do ambiente	Muito reduzido; apenas um corredor estreito observado	Moderado; arredores do campo marcados com boa coerência	Elevado; praticamente toda a área entre fileiras mapeada
Definição do espaço livre	Corredor linear contínuo, sem irregularidades	Estrutura poligonal bem definida	Corredores entre fileiras bem definidos e navegáveis
Identificação de obstáculos / vegetação	Não aplicável	Bordas externas claramente detetadas	Fileiras representadas como bandas densas e paralelas
Densidade de voxels ocupados	Muito baixa	Média	Alta (devido às videiras)
Densidade de voxels livres	Baixa-média	Média	Elevada entre fileiras; reduzida dentro das bandas de vegetação
Regiões desconhecidas	Muito elevadas	Moderadas	Reduzidas (exceto extremidades do mapa)
Ruído na reconstrução	Quase inexistente	Pequeno nas curvas	Pequeno a moderado nas extremidades das fileiras
Coerência espacial global	Excelente	Boa a muito boa	Muito boa; mapa consistente ao longo das 7 fileiras
Adequação para navegação	Adequado apenas para simulação n.º 1	Adequado e robusto	Excelente; ideal para navegação agrícola

Na Simulação n.º 1, correspondente ao seguimento de uma trajetória retilínea, o mapa final encontra-se representado na Figura 4-8. Verifica-se que o espaço livre

ao longo da linha de deslocamento é claramente identificado, formando um corredor contínuo que coincide com o percurso percorrido pelo sistema robótico. As regiões exteriores ao trajeto permanecem maioritariamente desconhecidas, o que é expectável dado que o campo de visão da câmara RGB-D abrange apenas o espaço imediatamente adjacente à trajetória.

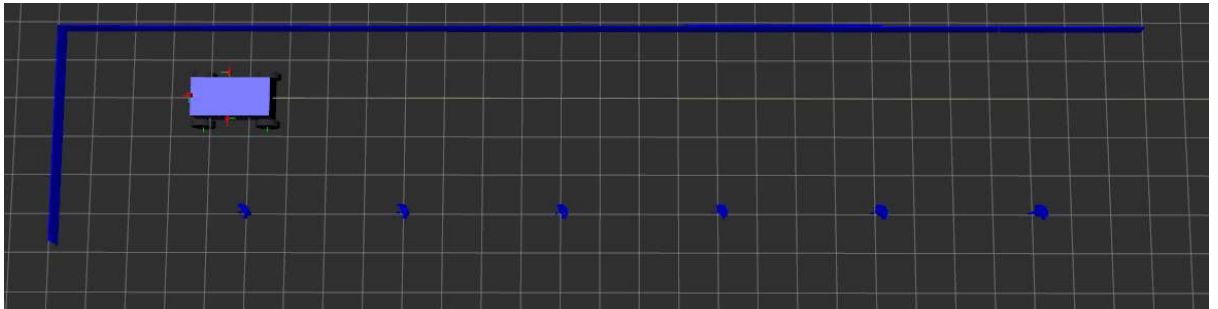


Figura 4-8 OctoMap na trajetória retilínea (simulação n.º 1)

Na Simulação n.º 2, onde o sistema robótico percorre o contorno completo do cenário, o mapa 3D obtido é ilustrado na Figura 4-9. A geometria do octree mostra uma representação coerente das fronteiras do campo, visíveis como faixas contínuas de voxels ocupados. A forma poligonal resultante coincide com a trajetória percorrida, revelando a capacidade do OctoMap para registar de forma precisa as margens do ambiente à medida que o sistema robótico o delimita.

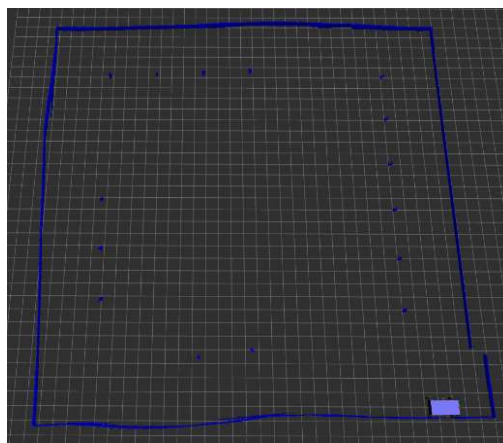


Figura 4-9 OctoMap na trajetória em torno do cenário (simulação n.º 2)

Nas mudanças de direção surgem pequenas irregularidades na superfície dos voxels, associadas ao ângulo oblíquo de incidência da câmara e à discretização a 5 cm, porém, estas irregularidades não comprometem a coerência global do mapa.

A Simulação n.º 3 constitui o ensaio mais representativo, uma vez que o sistema robótico percorre todas as fileiras de videiras, passando sucessivamente por corredores estreitos delimitados pela vegetação. O mapa final, apresentado na Figura 4.10, mostra duas características essenciais:

- Formação de faixas paralelas de voxels ocupados, correspondentes às fileiras de videiras;
- Existência de corredores centrais livres, representando corretamente o espaço navegável entre as plantas.

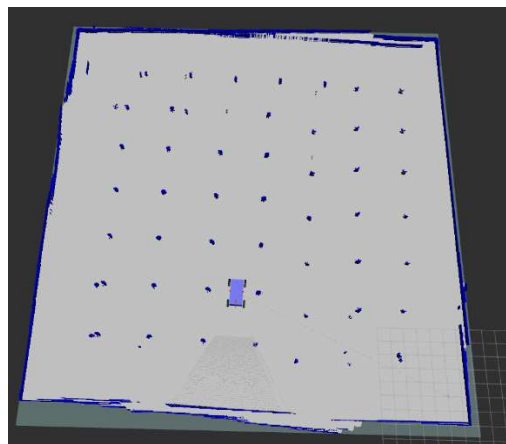


Figura 4-10 OctoMap na trajetória no cenário (simulação n.º 3)

À medida que o sistema robótico percorre cada fileira, a câmara RGB-D observa diretamente troncos e massas de vegetação, originando bandas densas de voxels ocupados ao longo de cada linha. Entre fileiras, observa-se a presença de espaços livres bem definidos, demonstrando que o OctoMap consegue distinguir de forma eficaz o espaço navegável da vegetação circundante. Em algumas zonas, nomeadamente nas extremidades das fileiras e em regiões observadas em ângulos desfavoráveis, surgem pequenas irregularidades de ocupação dispersa. Estas são consequências expectáveis da discretização volumétrica e do ruído de profundidade, mas não comprometem a utilidade do mapa para navegação.

A análise quantitativa apresentada na Tabela 4-5 mostra um aumento marcado do número de voxels ocupados relativamente às simulações anteriores, o que reflete corretamente a maior densidade de elementos vegetais observados. A fração de volume mapeado é superior, uma vez que o sistema robótico percorreu praticamente todo o espaço operacional entre as videiras.

O mapa resultante demonstra elevada consistência espacial, sendo adequado para a criação de mapas de custo estáveis utilizados pelo planeador local do NAV2.

De forma geral, as três simulações confirmam que o OctoMap consegue criar mapas tridimensionais coerentes com a geometria realista do ambiente simulado.

Nos cenários simples, o mapa é mais leve e pouco denso, mas em percursos estruturados (contorno) e em ambientes agrícolas densos (fileiras), a representação ajusta-se naturalmente à complexidade observada. A distinção entre zonas livres, ocupadas e desconhecidas mantém-se clara ao longo de todos os ensaios, evidenciando que o sistema de mapeamento tridimensional é suficientemente robusto para suportar tarefas de navegação e planeamento em contexto agrícola.

4.7 Discussão global

Os resultados obtidos nas simulações demonstram que a arquitetura desenvolvida que é composta por um modelo cinemático Ackermann, um controlador baseado num controlador convencional do tipo PID, o sistema da navegação NAV2 e o sistema de mapeamento volumétrico OctoMap, apresentam um desempenho consistente e tecnicamente robusto em ambiente simulado.

- Na Simulação n.º 1, o erro lateral permanece baixo e praticamente constante ao longo de toda a missão.
 - Este comportamento confirma que o controlador local *vector_pursuit_controller* consegue estabilizar a trajetória em

- situações onde o gradiente do custo é simples e onde a curvatura imposta é nula.
- O RMSE é reduzido e não se refletem as limitações impostas pelo modelo Ackermann, cuja capacidade de correção lateral é intrinsicamente dependente da orientação e da taxa de variação angular admissível.
 - A ausência de replaneamentos evidencia que o planeador global *Smac Planner* mantém uma solução estável e não necessita de reajustes ao longo do percurso.
- Na Simulação n.º 2, observa-se um aumento previsível do erro máximo nas zonas de mudança de direção, associado à curvatura abrupta nas zonas de alteração da direção.
 - Estes aumentos do erro são característicos de sistemas Ackermann e refletem as limitações geométricas da cinemática não-holonómica, especialmente quando a velocidade linear não é reduzida antes da manobra.
 - A rapidez de convergência após cada curva demonstra a eficácia combinada entre o controlador local e o modelo preditivo interno do *vector_pursuit_controller*, que ajusta dinamicamente o vetor de velocidades admissíveis com base na cinemática e dinâmica do robô.
 - A ausência de replaneamentos confirma que a trajetória global definida permanece válida ao longo de toda a missão.
 - A Simulação n.º 3, é a tarefa mais complexa, envolve a navegação sequencial em múltiplas fileiras de videiras.
 - O sistema opera repetidamente em corredores estreitos com margens de segurança reduzidas.

- O erro apresenta oscilações de amplitude superior às observadas na Simulação n.º 1, no entanto são dentro dos limites aceitáveis para operações agrícolas.
- As transições entre fileiras constituem pontos críticos, uma vez que exigem variações abruptas de orientação e ajustamentos contínuos do planeador local em zonas onde o mapa de custo apresenta descontinuidades nas extremidades das fileiras.
- O número reduzido de replaneamentos e a elevada percentagem de waypoints atingidos sugerem que o NAV2 consegue conjugar eficazmente as camadas global e local, preservando a missão sem perda de estabilidade.

Do ponto de vista do mapeamento, as três simulações confirmam que o OctoMap fornece uma representação volumétrica internamente consistente.

- Na Simulação n.º 1, o mapa restringe-se praticamente a um corredor estreito de voxels livres, com a maior parte do volume mantida como desconhecida. Esta característica é desejável, pois revela que o OctoMap não gera ocupação artificial em regiões não observadas.
- Na Simulação n.º 2, a octree regista com nitidez as fronteiras do cenário, evidenciando a capacidade do sistema para mapear feições com geometrias poligonais. Observam-se pequenas irregularidades nos limites, típicas da discretização e de leituras em ângulos oblíquos, mas que não comprometem a construção do costmap.
- A Simulação n.º 3 produz um mapa mais denso e informativo, com bandas contínuas de voxels ocupados que correspondem às videiras simuladas. A consistência destas bandas demonstra que o pipeline entre a câmara RGB-D e o OctoMap está corretamente calibrado. A persistência de regiões desconhecidas entre fileiras não visitadas confirma o funcionamento adequado dos limites superior e inferior de

probabilidade, impedindo a saturação das probabilidades e garantindo que a octree mantém a distinção essencial entre espaço livre, ocupado e não observado.

Em condições reais, é expectável um aumento do RMSE, da amplitude dos erros laterais e da inconsistência espacial nos mapas volumétricos. Contudo, a arquitetura ROS 2 utilizada foi estruturada de forma modular e extensível, permitindo substituir as fontes ideais do simulador por sensores reais (IMU, câmaras RGB-D, GNSS ou LiDAR), sem necessidade de reconfiguração estrutural da navegação.

Em síntese, os resultados evidenciam que a integração entre NAV2, modelo Ackermann e OctoMap é tecnicamente sólida e representa uma boa base funcional para a transição para um protótipo físico.

A arquitetura comporta-se de forma previsível, responde corretamente às restrições cinemáticas e gera mapas consistentes para suportar o planeamento.

5 CONCLUSÕES E TRABALHOS FUTUROS

Este projeto teve por objetivo desenvolver e avaliar uma arquitetura de navegação e controlo para um sistema robótico agrícola com cinemática do tipo Ackermann, baseada numa infraestrutura comum de software suportada pela plataforma ROS 2 que é válida tanto para ambientes simulados como reais.

O sistema robótico desenvolvido tem como finalidade trabalhar em campos agrícolas organizados em linhas de cultivo. Ao admitir-se a possibilidade de trabalhar sem sistemas de posicionamento absoluto fiáveis, desenvolveu-se uma solução integrada que combinasse modelação cinemática, perceção, mapeamento tridimensional e navegação autónoma.

Foi definido o modelo geométrico e cinemático do sistema robótico, recorrendo a uma representação em *links* e *joints* que descreve a estrutura do chassis, das rodas e dos sensores, bem como a relação entre os respetivos referenciais. Este modelo foi utilizado simultaneamente no simulador e na arquitetura destinada a um sistema robótico real, garantindo coerência entre os dois contextos e reduzindo o risco de divergência entre o comportamento previsto em ambiente virtual e o comportamento observado em campo.

A cinemática Ackermann foi escolhida para a descrição do movimento no plano, permitindo relacionar o ângulo de direção aplicado às rodas dianteiras, a velocidade de avanço e a evolução da pose do sistema robótico ao longo do tempo.

A cinemática constituiu o núcleo da odometria utilizada pela navegação, após a implementação da cinemática do sistema robótico foi desenvolvido uma camada de sensorização, para criar a perceção de todo o sistema, nesta integrou-se um IMU, uma câmara RGB-D, bem como um módulo de mapeamento tridimensional probabilístico baseado no “framework” OctoMap.

Com as medições de profundidade e da estimação de pose do sistema robótico, tornou-se possível construir mapas volumétricos que distinguem explicitamente entre espaço livre, ocupado e desconhecido, com resolução adaptativa e utilização eficiente de memória.

Estes mapas foram pensados para serem compatíveis com a navegação NAV2, possibilitando a alimentação de mapas de custo e algoritmos de planeamento de trajetórias em ambientes com obstáculos e estrutura tridimensional relevante, como é o caso de campos agrícolas com árvores, estruturas de suporte ou variações de relevo.

Em conjunto com os sensores e o modelo do sistema robótico foi implementada a arquitetura ROS 2 que interliga os nós com hierarquias bem definidas.

A hierarquia é construída da seguinte forma:

- Construção de referências;
- Localização;
- Mapeamento;
- Trajetória de referência;
- Interface no simulador e no RViz2;

Objetivo final é retratado por uma sequência de waypoints ou uma trajetória de referência, que o sistema robótico irá percorrer ao longo das linhas de cultivo.

A localização é obtida a partir da odometria cinemática fornecida pelo simulador e, numa fase futura, por sensores reais e algoritmos de fusão.

O mapeamento é assegurado pelo OctoMap, que integra continuamente observações da câmara de profundidade. A navegação é suportada pelo NAV2, que recebe objetivos espaciais, utilizando a odometria como fonte principal de pose e

que recorre ao mapa de ocupação para construir um mapa de custo, invocando assim os planeadores globais e locais para gerar trajetórias viáveis.

O controlo de movimento do sistema robótico é realizado em malha fechada, com controladores clássicos do tipo PID que ajustam a velocidade linear e o ângulo de direção em função do erro de seguimento de trajetória, respeitando as restrições impostas pela cinemática Ackermann.

A arquitetura definida é validada em ambiente de simulação, através de um conjunto de simulações representativas de operação agrícola.

Nas várias simulações o sistema robótico recebeu objetivos sob a forma de percursos ao longo das linhas, foram recolhidos dados de odometria, comandos de controlo e mapas OctoMap criados ao longo destes objetivos. A partir destes dados foram calculadas métricas de desempenho relacionadas com o erro de seguimento de trajetória, o tempo de missão, a percentagem de objetivos atingidos e a densidade e coerência do mapa de ocupação.

Os resultados obtidos indicam que nas simulações simples, a trajetória seguida pelo sistema robótico é idêntica à trajetória de referência, com erros médios compatíveis com aplicações de fertilização onde se admite alguma tolerância lateral relativamente ao centro da linha.

As maiores discrepâncias surgem tipicamente nas zonas de inversão de marcha e em curvas mais apertadas, o que é coerente com as limitações naturais da cinemática Ackermann e com a discretização das trajetórias planeadas.

No ponto de vista do mapeamento, os mapas OctoMap criados durante os ensaios reproduzem de forma coerente a geometria principal do ambiente simulado. Em cenários sem obstáculos, o espaço livre correspondente às linhas percorridas é claramente identificado, enquanto as regiões não visitadas permanecem na categoria de desconhecido, como seria de esperar.

Em cenários com obstáculos, estes surgem representados como regiões compactas de voxels ocupados, com forma e posição global aproximadamente coincidentes com os objetos existentes no mundo simulado.

Apesar de se verificar pequenas irregularidades em zonas de fronteira e em ângulos de observação menos favoráveis, associadas ao ruído e à discretização, o mapa obtido é suficientemente consistente para ser utilizado como base para mapas de custo e para suporte ao planeamento de trajetórias pela navegação.

A avaliação foi realizada apenas em simulação, utilizando odometria ideal e assumindo atuadores e sensores sem falhas significativas. Em contexto real, efeitos como derrapagem das rodas, irregularidades do terreno, atrasos de comunicação e ruído nos sensores irão, inevitavelmente, degradar o desempenho observado, exigindo calibração adicional, algoritmos de fusão mais sofisticados e uma análise mais aprofundada da robustez dos controladores e dos planeadores.

A modelação dinâmico-cinématica escolhida é relativamente simples, centrando-se na cinématica Ackermann e não considerando explicitamente limitações dinâmicas, como saturação de binário, transferência de carga ou perda de aderência em manobras mais agressivas.

As simulações embora representativas de algumas situações típicas, não captam toda a complexidade de um campo agrícola real, onde podem existir objetos dinâmicos, vegetação móvel, variações bruscas de condições e interferências de outros equipamentos.

Por fim, a integração do processo de fertilização foi tratada apenas a nível conceptual, sem implementação de um circuito de controlo dedicado à taxa de aplicação nem utilização de sensores de caudal, pelo que a ligação entre a navegação do sistema robótica e a qualidade agronómica da aplicação de fertilizante permanece em aberto.

Estas limitações definem, de forma bastante clara, as direções de um trabalho futuro. Uma evolução natural consiste em transferir a arquitetura desenvolvida para um protótipo físico, mantendo a organização em ROS 2 e substituindo progressivamente as fontes de dados ideais do simulador por medições reais de encoders, IMU, câmara RGB-D ou outros sensores adequados.

Essa transição irá implicar a calibração da odometria, a definição de modelos de ruído e a utilização de algoritmos de fusão de dados para melhorar a estimação de pose em condições adversas. Paralelamente, é desejável aprofundar o módulo de percepção e mapeamento, explorando variantes e extensões ao OctoMap, incluindo técnicas de aceleração em GPU, estratégias de remoção de objetos dinâmicos e, eventualmente, integração de informação semântica que permita distinguir diferentes tipos de obstáculos ou zonas de interesse agronómico.

Ao nível da navegação e controlo, uma linha de desenvolvimento interessante passa pela experimentação de controladores mais avançados para seguimento de trajetória, nomeadamente abordagens não lineares, baseadas em controlo ótimo ou em controlo preditivo, bem como pela afinação mais fina dos parâmetros do NAV2 para o contexto particular da robótica agrícola.

A introdução de algoritmos de exploração autónoma, capazes de determinar quais as zonas do campo que ainda não foram mapeadas ou fertilizadas e de planear percursos que maximizem a cobertura com o mínimo de recursos, constitui também uma extensão natural da lógica de missão utilizada neste trabalho.

Finalmente, a integração efetiva do controlo do processo de fertilização na arquitetura global, com a modelização do sistema de distribuição, medição da taxa de aplicação e implementação de um circuito de controlo que ajuste atuadores em função da velocidade e dos objetivos agronómicos. Permitirá aproximar o sistema de uma solução completa de agricultura de precisão, em que navegação, percepção e aplicação de insumos funcionam de forma verdadeiramente integrada.

Para concluir, o trabalho realizado mostra que é possível conceber uma arquitetura comum para sistema robótico real e simulado, assente em ROS 2, NAV2 e OctoMap, capaz de suportar navegação autónoma em cenários agrícolas estruturados.

A simulação mostra que a abordagem é tecnicamente viável e consistente, fornecendo uma base sólida sobre a qual podem ser construídas futuras extensões, tanto ao nível da implementação física como ao nível da sofisticação dos algoritmos de perceção, planeamento e controlo de processos.

A implementação do sistema de fertilização constitui a etapa funcional que permitirá completar o ciclo entre navegação autónoma, perceção do ambiente e aplicação de insumos no contexto agrícola. O sistema robótico deverá ser capaz de ativar e desativar o atuador de pulverização em função da sua posição, da fileira em que se encontra e das condições definidas pelo operador. Para esse efeito, prevê-se a integração de um módulo adicional de controlo da fertilização na arquitetura ROS 2 já desenvolvida.

A nível sensorial, o modelo do sistema robótico foi previamente equipado com dois sensores importantes para esta tarefa:

- Lidares laterais: utilizados como substitutos funcionais de sensores ultrassónicos reais, permitindo detetar a proximidade das videiras e manter o robô centrado no corredor.
- Sensores de partículas: configurados no simulador para representar o comportamento visual de uma pulverização. Estes sensores permitem visualizar o cone de aplicação do fertilizante, servindo como uma abstração gráfica da ação de pulverizar.

Apesar desta integração conceptual, o motor de simulação Gazebo apresenta limitações significativas no que diz respeito ao controlo dinâmico de fluxos de partículas. Em particular, o simulador não permite, de forma nativa e estável, ativar

ou desativar emissores de partículas por mensagens ROS2, o que impede a criação de um atuador de pulverização completamente funcional dentro do ambiente virtual.

Como consequência, a pulverização simulada pode ser representada visualmente, mas não pode ser controlada de modo programático em tempo real, limitando a capacidade de reproduzir com rigor o processo agrícola.

Numa implementação física futura, o módulo de fertilização seria responsável por:

- Receber sinais de comando, por exemplo, `/fertilizer_control/on` e `/fertilizer_control/off`),
- Supervisionar a posição do sistema robótico no mapa para aplicar fertilizante apenas nas fileiras desejadas,
- Ajustar a taxa de aplicação em função da velocidade de deslocação,
- Garantir a segurança, interrompendo a pulverização quando o sistema robótico sai do corredor.

Em ROS 2, estes comportamentos podem ser implementados através de um nó dedicado de controlo da pulverização, integrando sensores laterais para verificação de presença de vegetação e sincronizando a ativação com o modo de navegação. No mundo real, o atuador corresponderia a uma válvula solenoide ou bomba de pressão controlada via microcontrolador ou interface CAN.

Assim, embora o simulador não permita, nesta fase, testar de forma completamente funcional o processo de pulverização, a arquitetura do sistema robótico foi preparada de modo modular para que a transição para um protótipo físico seja direta. O nó de fertilização pode ser posteriormente acoplado à estrutura existente, explorando os mesmos princípios de perceção e controlo utilizados na navegação.

2

REFERÊNCIAS

- [1] Moshayedi, A. J., Khan, A. S., Yang, Y., Hu, J., & Kolahdooz, A. (2024). Robots in Agriculture: Revolutionizing Farming Practices. *EAI Endorsed Transactions on AI and Robotics*, 3. <https://doi.org/10.4108/airo.5855>
- [2] Hackenhaar, N. M., Hackenhaar, C., & Abreu, Y. V. de. (2015). Robótica na agricultura. *Interações*, 16(1), 119–129.
- [3] Backman, J., Oksanen, T., & Visala, A. (2012). Navigation system for agricultural machines: Nonlinear Model Predictive path tracking. *Computers and Electronics in Agriculture*, 82, 32–43. <https://doi.org/10.1016/j.compag.2011.12.009>
- [4] Albiero, D., Garcia, A. P., Umezu, C. K., & de Paulo, R. L. (2021). Swarm Robots in Agriculture. *arXiv*. <https://doi.org/10.48550/arXiv.2103.06732>
- [5] Wijesundara et al., 2023 – Accurate Crop Spraying with RTK and Machine Learning on an Autonomous Field Robot (ArXiv)
- [6] Hiremath et al., 2014 – Laser Range Finder Model for Autonomous Navigation of a Robot in a Maize Field Using a Particle Filter
- [7] Hasan et al., 2022 – Towards Technological Adaptation of Advanced Farming through AI, IoT, and Robotics (ArXiv)
- [8] M. Ayaz, M. Ammad-Uddin, Z. Sharif, A. Mansour and E. -H. M. Aggoune, "Internet-of-Things (IoT)-Based Smart Agriculture: Toward Making the Fields Talk," in *IEEE Access*, vol. 7, pp. 129551-129583, 2019, doi: 10.1109/ACCESS.2019.2932609.
- [9] Vougioukas, S. (2019) – Agricultural Robotics for Planting and Seeding Automation. *Annual Review of Control*, 53, 1-15. DOI: 10.1016/j.arcontrol.2019.02.003.

- [10] Aguiar, A. S., Santos, F. N., Cunha, J. B., Sobreira, H., & Sousa, A. J. (2020) – Localization and Mapping for Robots in Agriculture and Forestry: A Survey. *Robotics*, 9(4), 97. DOI: 10.3390/robotics9040097.
- [11] Droukas, L., Doulgeri, Z., Tsakiridis, N. L., Triantafyllou, D., et al. (2023) – A Survey of Robotic Harvesting Systems and Enabling Technologies. *Journal of Intelligent & Robotic Systems*, 107, 21. DOI: 10.1007/s10846-022-01793-z.
- [12] Raussendorf, Fruit robot (2020), <https://www.raussendorf.de/en/fruit-robot.html>.
- [13] P.Makers, Greenbot (2020)<https://www.precisionmakers.com/en/greenbot-2/>.
- [14] Bacenetti, J., Paleari, L., Tartarini, S., Vesely, F. M., Foi, M., Movedi, E., Ravasi, R. A., Bellopede, V., Durello, S., Ceravolo, C., Amicizia, F., & Confalonieri, R. (2020). May smart technologies reduce the environmental impact of nitrogen fertilization. A case study for paddy rice. *Science of the Total Environment*, 715, 136956.
- [15] Nowaki, R. H. D., Parent, S.-É., Cecílio Filho, A. B., Rozane, D. E., Meneses, N. B., dos Santos da Silva, J. A., Natale, W., & Parent, L. E. (2017). Phosphorus over-fertilization and nutrient misbalance of irrigated tomato crops in Brazil. *Frontiers in Plant Science*, 8, 825. <https://doi.org/10.3389/fpls.2017.00825>
- [16] Francis J. Pierce, Peter Nowak, Aspects of Precision Agriculture, *Advances in Agronomy*, Volume 67, 1999, Pages 1-85, ISSN 0065-2113, ISBN 9780120007677, [https://doi.org/10.1016/S0065-2113\(08\)60513-1](https://doi.org/10.1016/S0065-2113(08)60513-1).
- [17] P. C. Robert, Precision agriculture: A challenge for crop nutrition management, November 2002, *Plant and Soil* 247(1):143-149, DOI: 10.1023/A:1021171514148

- [18] Dag, Arnon & Erel, Ran & Kerem, Zohar & Ben-Gal, Alon & Stern, N. & Bustan, A. & Zipori, I. & Yermiyahu, Uri. (2018). Effect of nitrogen availability on olive oil quality. *Acta Horticulturae*. 465-470. 10.17660/ActaHortic.2018.1199.74.
- [19] N. Schor, A. Bechar, T. Ignat, A. Dombrovsky, Y. Elad e S. Berman, “Robotic Disease Detection in Greenhouses: Combined Detection of Powdery Mildew and Tomato Spotted Wilt Virus,” *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 354–360, 2016, doi: 10.1109/LRA.2016.2518214.
- [20] G. Adamides, C. Katsanos, I. Constantinou, G. Christou, M. Xenos, T. Hadzilacos e Y. Edan, “Design and development of a semi-autonomous agricultural vineyard sprayer: Human–robot interaction aspects,” *Journal of Field Robotics*, vol. 34, no. 8, pp. 1407–1426, 2017, doi: 10.1002/rob.21721.
- [21] D. Turner, A. Lucieer and C. Watson, Development of an unmanned aerial vehicle (uav) for hyper-resolution vineyard mapping based on visible, multispectral and thermal imagery *Proceedings of 34th International Symposium on Remote Sensing of Environment* 101 2011.
- [23] J. Torres-Sánchez, J. M. Peña, A. I. de Castro e F. López-Granados, “Multi-temporal mapping of the vegetation fraction in early-season wheat fields using images from UAV,” *Computers and Electronics in Agriculture*, vol. 103, pp. 104–113, abr. 2014, doi: 10.1016/j.compag.2014.02.009.
- [24] J. Kim, S. Kim, C. Ju e H. I. Son, “Unmanned aerial vehicles in agriculture: A review of perspective of platform, control, and applications,” *IEEE Access*, vol. 7, pp. 105100–105115, 2019, doi: 10.1109/ACCESS.2019.2932119.
- [25] S. Meivel, K. Dinakaran, N. Gandhiraj and M. Srinivasan, Remote sensing for urea spraying agricultural (uav) system, in 2016 3rd International Conference on Advanced Computing and Communication Systems (ICACCS), 2016.

- [26] X. Wu, S. Aravecchia, P. Lottes, C. Stachniss e C. Pradalier, “Robotic weed control using automated weed and crop classification,” *Journal of Field Robotics*, vol. 37, no. 2, pp. 322–340, 2020, doi: 10.1002/rob.21938.
- [27] Bac, C. W., Hemming, J., & Van Henten, E. J. (2014). Harvesting robots for high-value crops: State-of-the-art review and challenges ahead. *Journal of Field Robotics*, 31(6), 888-911.
- [28] Baeten, Johan & Donn e, Kevin & Boedrij, Sven & Beckers, Wim & Claesen, Eric. (2007). *Autonomous Fruit Picking Machine: A Robotic Apple Harvester*. Springer Tracts in Advanced Robotics. 42. 10.1007/978-3-540-75404-6_51.
- [29] Fern andez, R., Rodr guez, F., Cerrada, M., & Benavides, C. (2018). Design and development of a robotic system for harvesting tomatoes. *Robotics and Autonomous Systems*, 107, 42-51.
- [30] Henten, E. J., Hemming, J., Van Tuijl, B. A. J., Kornet, J. G., Meuleman, J., Bontsema, J., & Van Os, E. A. (2003). An autonomous robot for harvesting cucumbers in greenhouses. *Autonomous Robots*, 13(3), 241-258.
- [31] Lehnert, C., English, A., McCool, C., Tow, A., & Perez, T. (2017). Autonomous sweet pepper harvesting for protected cropping systems. *IEEE Robotics and Automation Letters*, 2(2), 872-879.
- [32] Silwal, A., Davidson, J. R., Karkee, M., Mo, C., Zhang, Q., & Lewis, K. (2017). Design and field evaluation of a robotic apple harvester. *Computers and Electronics in Agriculture*, 142, 351-361.
- [33] Van Henten, E. J., Bac, C. W., & Hemming, J. (2019). Trends in robotics for horticulture. *Acta Horticulturae*, 1235, 31-38.
- [34] C. W. Bac, E. J. van Henten, J. Hemming e Y. Edan, “Harvesting Robots for High-value Crops: State-of-the-art Review and Challenges Ahead,” *Journal of Field Robotics*, vol. 31, no. 6, pp. 888–911, 2014, doi: 10.1002/rob.21525. cris.bgu.ac.il+1

- [35] Y. Ge, Y. Xiong, G. L. Tenorio e P. J. From, “Fruit Localization and Environment Perception for Strawberry Harvesting Robots,” *IEEE Access*, vol. 7, pp. 147642–147652, 2019, doi: 10.1109/ACCESS.2019.2946369. repositorio.inesctec.pt+1
- [36] S. Birrell, J. Hughes, J. Y. Cai e F. Iida, “A field-tested robotic harvesting system for iceberg lettuce,” *Journal of Field Robotics*, vol. 37, no. 2, pp. 225–245, 2020, doi: 10.1002/rob.21888.
- [37] Slaughter, D. C., Giles, D. K., & Downey, D. (2008). Autonomous robotic weed control systems: A review. *Computers and Electronics in Agriculture*, 61(1), 63-78.
- [38] Zhang, Q., Chen, M. E. S., & Li, B. (2017). A visual navigation algorithm for paddy field weeding robot based on image understanding. *Computers and Electronics in Agriculture*, 143, 66-78.
- [39] Pérez-Ruiz, M., Slaughter, D. C., Gliever, C. J., & Upadhyaya, S. K. (2014). Automatic GPS-based guidance system for agricultural robots. *Precision Agriculture*, 15(6), 621-635.
- [40] Choi, K. H., Han, S. K., Han, S. H., Park, K. H., Kim, K. S., & Kim, S. (2015). Morphology-based guidance line extraction for an autonomous weeding robot in paddy fields. *Computers and Electronics in Agriculture*, 113, 266-274.
- [41] N.D. Tillett, T. Hague, A.C. Grundy, A.P. Dedousis, Mechanical within-row weed control for transplanted crops using computer vision, *Biosystems Engineering*, Volume 99, Issue 2,2008, Pages 171-178, ISSN 1537-5110, <https://doi.org/10.1016/j.biosystemseng.2007.09.026>.
- [42] Raffik, R., Abishek, D., Mayukha, S., Tharun, R., Hemchander, J., & Deepakumar, S. (2021). Autonomous Weeding Robot for Organic Farming Fields. *IEEE International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)*.

- [43] Fennimore, S. A., Slaughter, D. C., Siemens, M. C., Leon, R. G., & Saber, M. N. (2016). Technology for automation of weed control in specialty crops. *Weed Technology*, 30(3), 560-567.
- [44] F. N. dos Santos, H. Sobreira, D. Campos, R. Morais, A. P. Moreira e O. Contente, “Towards a Reliable Robot for Steep Slope Vineyards Monitoring,” *Journal of Intelligent & Robotic Systems*, vol. 83, pp. 429–444, 2016, doi: 10.1007/s10846-016-0340-5
- [45] Grisso, R., Kocher, M., Vaughan, D. (2010). *Tractor Selection and Operation*. Virginia Cooperative Extension.
- [46] Kutzbach, H. D. (2000). Trends in Power and Machinery. *Journal of Agricultural Engineering Research*, 76(3), 237–247.
- [47] S. Birrell, J. Hughes, J. Y. Cai and F. Iida, *Journal of Field Robotics* 37, 225 (2020).
- [48] Pedersen, S. M., Fountas, S., Have, H., & Blackmore, B. S. (2006). Agricultural robots—system analysis and economic feasibility. *Precision Agriculture*, 7, 295–308.
- [49] Shalaby, M., El-Ashmawy, M., & Saady, H. (2021). Recent Trends in the Development of Tractors for Sustainable Agriculture. *Sustainability*, 13(12), 6650. <https://doi.org/10.3390/su13126650>
- [50] López-García, R., Peña, J. M., & Andújar, J. M. (2023). Smart tractors and soil health: challenges and perspectives. *Agricultural Systems*, 203, 103552. <https://doi.org/10.1016/j.agsy.2022.103552>
- [51] Marcolin, E., Zortea, D. C., & Chiumenti, A. (2022). Electrification in agricultural machinery: Current status and future trends. *Renewable and Sustainable Energy Reviews*, 161, 112360. <https://doi.org/10.1016/j.rser.2022.112360>

- [52] González, S., Ruiz, R., & Pérez, M. (2023). The role of autonomous tracked vehicles in precision farming: A case study in soil management. *Computers and Electronics in Agriculture*, 204, 107511.
- [53] Martins, C., Pimentel, A., & Figueiredo, R. (2022). Tracked vehicles in agricultural production: A technological overview. *Technology in Agriculture Journal*, 31(3), 115–128.
- [54] S. Bargoti e J. P. Underwood, “Image segmentation for fruit detection and yield estimation in apple orchards,” *Journal of Field Robotics*, vol. 34, no. 6, pp. 1039–1060, 2017, doi: 10.1002/rob.21699.
- [55] Jazar, R. N. (2014). *Vehicle Dynamics: Theory and Application*. Springer.
- [56] Rill, G. (2011). *Road Vehicle Dynamics: Fundamentals and Modeling*. CRC Press.
- [57] Gillespie, T. D. (1992). *Fundamentals of Vehicle Dynamics*. SAE International.
- [58] P. Gautam, S. Sahai, S. S. Kelkar, P. S. Agrawal e M. Reddy D, “Designing Variable Ackerman Steering Geometry for Formula Student Race Car,” *International Journal of Analytical, Experimental and Finite Element Analysis*, vol. 8, no. 1, pp. 1–11, 2021, doi: 10.26706/ijaefea.1.8.20210101..
- [59] Weinstein, A. J., & Moore, K. L. "Pose Estimation of Ackerman Steering Vehicles for Outdoors Autonomous Navigation." *IEEE*, 2010.
- [60] Carpio, R. F., Potena, C., Maiolini, J., Ulivi, G., Rosselló, N. B., Garone, E., & Gasparri, A. (2020). A Navigation Architecture for Ackermann Vehicles in Precision Farming. *IEEE Robotics and Automation Letters*, 5(2), 1103–1110. doi:10.1109/LRA.2020.2967306.
- [61] Bascetta, L., Cucci, D. A., & Matteucci, M. (2016). Kinematic trajectory tracking controller for an all-terrain Ackermann steering vehicle. *IFAC-PapersOnLine*, 49(15), 13–18. doi:10.1016/j.ifacol.2016.07.600.

- [62] Y.-H. Kang, D.-C. Pang e Y.-C. Zeng, “Optimal Dimensional Synthesis of Ackermann Steering Mechanisms for Three-Axle, Six-Wheeled Vehicles,” *Applied Sciences*, vol. 15, no. 2, art. 800, 2025, doi: 10.3390/app15020800.
- [63] G. AĞAKIŞI e F. ÖZTÜRK, “Kinematics & Compliance Validation of a Vehicle Suspension and Steering Kinematics Optimization Using Neural Networks,” *Mechanika*, vol. 29, no. 3, pp. 243–251, 2023, doi: 10.5755/j02.mech.31983.
- [64] Bevly, D. M., & Parkinson, B. W. "Integrated GPS/INS Systems for Land Vehicles." *Journal of Navigation*, 2007.
- [65] J. Diebel, “Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors,” *Matrix*, vol. 58, p. 15, 2006.
- [66] N. A. I. Ruslan, N. H. Amer, K. Hudha, Z. A. Kadir, S. A. F. M. Ishak, e S. M. F. S. Dardin, «Modelling and control strategies in path tracking control for autonomous tracked vehicles: A review of state of the art and challenges», *Journal of Terramechanics*, vol. 105, pp. 67–79, fev. 2023, doi: 10.1016/j.jterra.2022.10.003.
- [67] Indirect Adaptive PID Control Using Neural Network and Discrete Extended Kalman Filter For Mobile Robot Control, Mohammed Yousri Silaa, Aissa Bencherif and Oscar Barambones
- [68] 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].
- [69] Chen, Z., Wang, H., & Li, Y. (2021). Vision and IMU fusion for autonomous navigation of agricultural robots in GPS-denied environments. *Computers and Electronics in Agriculture*, 181, 105946.
- [70] Li, X., Yang, Y., & Zhang, L. (2022). LiDAR and IMU-based navigation for autonomous systems in agricultural applications. *Sensors*, 22(5), 1789.

- [71] Wang, Q., Li, S., & Zhang, Y. (2021). Application of IMU and GPS fusion in agricultural robot navigation. *Journal of Field Robotics*, 38(9), 1250-1263.
- [72] Miranda, A. C. (2022). Teaching ROS2 with a Minimalistic Mobile Robot. Universidade do Porto.
- [73] L. Osborn, “This or That: RGB vs. Multispectral Sensors,” 2023 Janeiro 18. [Online]. [Acedido em Julho 2023].
- [74] ZHMUD, V. A.; KONDRATIEV, N. O.; KUZNETSOV, K. A.; TRUBIN, V. G.; DIMITROV, L. V. Application of ultrasonic sensor for measuring distances in robotics. *Journal of Physics: Conference Series*, [S. l.], v. 1015, n. 3, p. 032189, maio 2018. DOI: 10.1088/1742-6596/1015/3/032189.
- [75] Yuya Maruyama, Shinpei Kato, and Takuya Azumi. 2016. Exploring the performance of ROS2. In *Proceedings of the 13th International Conference on Embedded Software (EMSOFT '16)*. Association for Computing Machinery, New York, NY, USA, Article 5, 1–10. <https://doi.org/10.1145/2968478.2968502>
- [76] Castilho, J. P. C. (2022). ROS 2.0 – Study and Evaluation of ROS 2 in comparison with ROS 1. Universidade de Coimbra.
- [77] Macenski, S., Soragna, A., Carroll, M., & Ge, Z. (2023). Impact of ROS 2 Node Composition in Robotic Systems.
- [78] Chovet, L. P., Garcia, G. M., Bera, A., Richard, A., Yoshida, K., & Olivares-Mendez, M. A. (2024). Evaluation of ROS 2 Middleware in Mesh Networks for Space Robots.
- [79] Lienen, C., Nowosad, A. P., & Platzner, M. (2023). Enhancing Robot Communication with Gateways.
- [80] Jiang, L., Li, H., & Zhang, C. (2020). Visual Odometry using Convolutional Neural Networks for Real-time Applications. *IEEE Transactions on Robotics*, 36(4), 1123-1135.

- [81] Zhang, Y., Wang, H., & Li, X. (2021). Visual-Inertial Odometry for Autonomous Navigation in Dynamic Environments. *Sensors*, 21(8), 2689.
- [82] Sundaram, R., Lee, J., & Barriuso, J. (2022). Real-time Monocular Visual Odometry for High-Speed Autonomous Vehicles. *Journal of Field Robotics*, 39(5), 722-735.
- [83] Li, S., Wang, X., & Liu, H. (2022). Robust Monocular Visual Odometry Using Deep Learning Techniques. *IEEE Transactions on Intelligent Vehicles*, 7(1), 49-61.
- [84] Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., ... & Leonard, J. J. (2016). "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age." *IEEE Transactions on Robotics*, 32(6), 1309-1332.
- [85] Davison, A. J., Reid, I. D., Molton, N. D., & Stasse, O. (2007). "MonoSLAM: Real-time single camera SLAM." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6), 1052-1067.
- [86] Engel, J., Koltun, V., & Cremers, D. (2014). "Direct sparse odometry." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3), 611-625.
- [87] Mur-Artal, R., & Tardós, J. D. (2017). "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras." *IEEE Transactions on Robotics*, 33(5), 1255-1262.
- [88] Wang, X., Clark, R., Wen, H., & Trigoni, N. (2017). "DeepVO: Towards end-to-end visual odometry with deep Recurrent Convolutional Neural Networks." *IEEE International Conference on Robotics and Automation (ICRA)*, 2043-2050.
- [89] Kannapiran, S., Bendapudi, N., Yu, M.-Y., Parikh, D., Berman, S., Vora, A., & Pandey, G., "Stereo Visual Odometry with Deep Learning-Based Point and Line Feature Matching using an Attention Graph Neural Network," *arXiv preprint arXiv:2308.01125*, 2023.

- [90] CAMPOS, C., ELHROUSSI, M., GARCÍA, J. & TARDÓS, J. (2021). ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multi-Map SLAM. *IEEE Transactions on Robotics*, 37(6), 1874-1890.
- [91] HUANG, Z., LIU, F., ZHANG, Y. & WANG, J. (2023). GNSS-Vision-Inertial SLAM for Agricultural Robotics: A Robust Approach for Outdoor Localization. *Sensors*, 23(4), 1256.
- [92] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, W. Burgard, “OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees,” *Autonomous Robots*, vol. 34, pp. 189–206, 2013.
- [93] OctoMap Project, “OctoMap – 3D occupancy mapping,” documentação online, octomap.github.io, acessado em 2025.
- [94] H. Min, K. M. Han, Y. J. Kim, “OctoMap-RT: Fast Probabilistic Volumetric Mapping Using Ray-Tracing GPUs,” *IEEE Robotics and Automation Letters*, vol. 8, no. 9, pp. 5696–5703, 2023.
- [95] Z. Liu, P. van Oosterom, J. Balado, et al., “Data Frame Aware Optimized Octomap-Based Dynamic Object Detection and Removal in Mobile Laser Scanning Data,” *Alexandria Engineering Journal*, vol. 64, 2023.
- [96] J. Hou, et al., “Octree-Based Approach for Real-Time 3D Indoor Mapping Using a Handheld RGB-Depth Camera,” *ISPRS Archives*, vol. XLVIII-1/W1-2023, pp. 183–190, 2023.
- [97] D. Čapek, *Volumetric Mapping Onboard Unmanned Helicopter*, Bachelor Thesis, Czech Technical University in Prague, 2023.
- [98] M. Basso, D. Stocchero, P. Schnarndorf, E. P. de Freitas, “Merging Three-Dimensional Occupancy Grid Maps to Support Multi-UAVs Cooperative Navigation Systems,” *Journal of Field Robotics*, 2023.

[99] P. Chen, et al., “OctoCache: Caching Voxels for Accelerating 3D Occupancy Mapping,” Proceedings of ASPLOS, 2025.

[100] H. Li, et al., “An Autonomous Navigation Method for Orchard Mobile Robots Based on Octree-Optimized 3D Point Clouds,” Frontiers in Plant Science, 2024/2025.