

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

CAIO HURTADO SORDI

LUCAS CORADIN RECH

**PROPOSTA DE UMA ARQUITETURA DE POSICIONAMENTO VISUAL PARA
APLICAÇÕES EM VANTS AUTÔNOMOS COLABORATIVOS**

CURITIBA

2022

**CAIO HURTADO SORDI
LUCAS CORADIN RECH**

**PROPOSTA DE UMA ARQUITETURA DE POSICIONAMENTO VISUAL PARA
APLICAÇÕES EM VANTS AUTÔNOMOS COLABORATIVOS**

**Proposal of a visual positioning architecture for collaborative autonomous
UAV applications**

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia de Computação do Curso de Bacharelado em Engenharia de Computação da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Marco Aurélio Wehrmeister

Coorientador: Prof. Dr. Alvaro Rogério Cantieri

CURITIBA

2022



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

**CAIO HURTADO SORDI
LUCAS CORADIN RECH**

**PROPOSTA DE UMA ARQUITETURA DE POSICIONAMENTO VISUAL PARA
APLICAÇÕES EM VANTS AUTÔNOMOS COLABORATIVOS**

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia de Computação do Curso de Bacharelado em Engenharia de Computação da Universidade Tecnológica Federal do Paraná.

Data de aprovação: 24/agosto/2022

João Alberto Fabro
Doutorado
Universidade Tecnológica Federal do Paraná

Eduardo Todt
Doutorado
Universidade Federal do Paraná

Marco Aurélio Wehrmeister
Doutorado
Universidade Tecnológica Federal do Paraná

**CURITIBA
2022**

RESUMO

Os VANTs autônomos oferecem vantagens em aplicações industriais, agrícolas, de inspeção ambiental e logística. Às vezes, o uso de VANTs cooperativos é importante para solucionar demandas específicas ou obter ganho de produtividade nessas aplicações. Um importante desafio técnico é o posicionamento preciso entre dois ou mais VANTs em um voo de tarefa cooperativa. Algumas técnicas fornecem soluções, como o posicionamento GNSS, visual e LIDAR SLAM, e algoritmos inteligentes de visão computacional, mas todas essas técnicas apresentam limitações que devem ser resolvidas para funcionar adequadamente em ambientes específicos. A proposta de novos métodos de posicionamento cooperativo é importante para enfrentar esses desafios. O presente trabalho propõe uma avaliação de uma arquitetura de posicionamento relativo visual entre duas pequenas aeronaves multi-rotores VANT trabalhando em uma operação mestre-escravo, baseada em uma ferramenta de etiqueta de Realidade Aumentada. Os resultados da simulação obtiveram medidas de erro inferiores a 0,2cm de média e 0,01 desvio padrão para as direções X, Y e Z. Os experimentos do mundo real executando voo autônomo com o VANT servo comandado pelo VANT mestre obtiveram sucesso em 8 de 10 rodadas de experimentos, indicando que a arquitetura proposta é uma boa abordagem para a construção de aplicações utilizando VANTs cooperativos.

Palavras-chave: drone; robica; vant; posicionamento visual; cooperativa.

ABSTRACT

Autonomous UAVs offer advantages in industrial, agriculture, environment inspection, and logistics applications. Sometimes using cooperative UAVs is important to solve specific demands or achieve productivity gains in these applications. An important technical challenge is the precise positioning between two or more UAVs in a cooperative task flight. Some techniques provide solutions, e.g. the GNSS positioning, visual and LIDAR SLAM, and computer vision intelligent algorithms, but all these techniques present limitations that must be solved to work properly in specific environments. The proposal of new cooperative position methods is important to face these challenges. The present work proposes an evaluation of a visual relative positioning architecture between two small UAV multi-rotor aircrafts working in a master-slave operation, based on an Augmented Reality tag tool. The simulation results showed absolute error measurements lower than 0.2cm in average and 0.01 standard deviation for X, Y and Z directions. The real-world experiments executing flight with the slave UAV commanded by the master UAV achieved success in 8 out of 10 experiment rounds, indicating that the proposed architecture is a good approach to build cooperative master-slave UAV applications.

Keywords: drone; robotic; uav; visual positioning; cooperative.

LISTA DE FIGURAS

Figura 1 – Exemplo de VANT de asa fixa.	16
Figura 2 – Exemplo de VANT multirroto de asa rotativa.	16
Figura 3 – Eixos de movimentação e rotação de um VANT.	17
Figura 4 – Etiquetas utilizadas pelo algoritmo Alvar.	18
Figura 5 – Diagrama de um controlador de altitude usado em VANTS.	19
Figura 6 – Gráficos representando o comportamento do sinal de resposta ao longo do tempo para valores diferentes de K_P . a) K_P alto, resposta rápida e comportamento oscilatório. b) K_P ideal. c) K_P baixo, resposta lenta.	20
Figura 7 – Exemplo de ambiente desenvolvido no CoppeliaSim.	22
Figura 8 – Arquitetura simplificada do sistema.	23
Figura 9 – Diagrama do sistema simulado com um VANT.	24
Figura 10 – Grafo dos nós ROS.	24
Figura 11 – Primeira cena desenvolvida no CoppeliaSim.	25
Figura 12 – Diagrama do sistema simulado com múltiplos VANTS escravos.	26
Figura 13 – Cena desenvolvida no CoppeliaSim com múltiplos VANTS escravos.	26
Figura 14 – Experimento realizado com apenas um VANT real.	27
Figura 15 – Diagrama do sistema elaborado para teste com um VANT.	28
Figura 16 – Grafo dos nós do ROS.	28
Figura 17 – VANTS utilizados nos experimentos reais.	29
Figura 18 – Diagrama do sistema elaborado para teste com dois VANTS.	29
Figura 19 – Gráficos da posição do VANT escravo em relação ao VANT mestre. Imagens ampliadas nos Anexos A, B e C	31
Figura 20 – Gráficos da posição dos eixos X e Y do VANT escravo em relação ao VANT mestre. Imagens ampliadas nos Anexos D, E e F	31
Figura 21 – Rota do VANT escravo seguindo o VANT mestre.	32
Figura 22 – VANTS escravos voando guiados pelo VANT mestre.	33
Figura 23 – Gráficos da posição dos VANTS escravos guiados pelo VANT mestre. Imagens ampliadas nos Anexos G, H, I e J	34
Figura 24 – Visão da câmera.	36

Figura 25 – Rota realizada pelo VANT. Imagem ampliada no Anexo K	38
Figura 26 – Gráficos da posição do VANT. Imagens ampliadas nos Anexos L e M . .	38
Figura 27 – Imagens do experimeto da arquitetura com VANT mestre e escravo. . .	39
Figura 28 – Gráficos da posição do VANT escravo em relação ao mestre. Imagens ampliadas nos Anexos N, O e P	40

LISTA DE TABELAS

Tabela 1 – Erros das medidas nas três rotas.	32
Tabela 2 – Erros de medida das etiquetas dos múltiplos VANTs escravos.	35
Tabela 3 – Posições em que os pontos foram fixados.	35
Tabela 4 – Média e desvio padrão das medidas.	36
Tabela 5 – Raiz quadrado do erro médio das medidas.	37
Tabela 6 – Pontos da rota planejada.	37

LISTA DE ABREVIATURAS E SIGLAS

Siglas

VANT	Veículo aéreo não tripulado
GPS	<i>Global Positioning System</i> , do inglês, sistema de posicionamento global
ROS	<i>Robot Operating System</i> , do inglês, sistema operativo de robôs
CSV	<i>Comma separated values</i> , do inglês, valores separados por vírgula. São arquivos de texto que fazem uma ordenação de bytes ou um formato de terminador de linha, separando valores com vírgulas
USB	<i>Universal Serial Bus</i> , do inglês, porta serial universal

SUMÁRIO

1	INTRODUÇÃO	10
1.1	Motivação	10
1.2	Objetivos	11
1.2.1	Objetivos Gerais	11
1.2.2	Objetivos Específicos	11
2	REVISÃO DE LITERATURA	12
3	FUNDAMENTAÇÃO TEÓRICA	14
3.1	ROBÓTICA MÓVEL	14
3.2	VEÍCULO AÉREO NÃO TRIPULADO	15
3.3	VISÃO COMPUTACIONAL	17
3.4	PROCESSAMENTO DIGITAL DE IMAGENS	18
3.5	RASTREAMENTO BASEADO EM MARCADORES	18
3.6	CONTROLADORES PID	18
3.6.1	Controlador Proporcional	19
3.6.2	Controlador Integral	19
3.6.3	Controlador Derivativo	20
3.6.4	Algoritmo Ziegler–Nichols	20
3.7	SIMULAÇÃO ROBÓTICA	21
3.7.1	CoppeliaSim	21
3.8	<i>ROBOT OPERATING SYSTEM</i>	22
4	ARQUITETURA DO SISTEMA	23
4.1	SIMULAÇÃO	23
4.2	IMPLEMENTAÇÃO NO MUNDO REAL	26
5	EXPERIMENTOS E RESULTADOS	30
5.1	AMBIENTE SIMULADO	30
5.1.1	Experimento com um VANT escravo	30
5.1.2	Experimento com múltiplos VANTs escravos	31
5.2	AMBIENTE REAL	33
5.2.1	Aferição do erro de medida das etiquetas	33
5.2.2	Experimento com um VANT escravo e a câmera fixa	35

5.2.3	Experimento com um VANT mestre e um VANT escravo	38
6	CONCLUSÕES	41
6.1	TRABALHOS FUTUROS	42
	REFERÊNCIAS	43
	ANEXO A GRÁFICO DA POSIÇÃO DO VANT FIGURA 19 (A)	47
	ANEXO B GRÁFICO DA POSIÇÃO DO VANT FIGURA 19 (B)	49
	ANEXO C GRÁFICO DA POSIÇÃO DO VANT FIGURA 19 (C)	51
	ANEXO D GRÁFICO DA POSIÇÃO DO VANT FIGURA 20 (A)	53
	ANEXO E GRÁFICO DA POSIÇÃO DO VANT FIGURA 20 (B)	55
	ANEXO F GRÁFICO DA POSIÇÃO DO VANT FIGURA 20 (C)	57
	ANEXO G GRÁFICOS DA POSIÇÃO DOS VANTS ESCRAVOS GUIA- DOS PELO VANT MESTRE FIGURA 23 (A)	59
	ANEXO H GRÁFICOS DA POSIÇÃO DOS VANTS ESCRAVOS GUIA- DOS PELO VANT MESTRE FIGURA 23 (B)	61
	ANEXO I GRÁFICOS DA POSIÇÃO DOS VANTS ESCRAVOS GUIA- DOS PELO VANT MESTRE FIGURA 23 (C)	63
	ANEXO J GRÁFICOS DA POSIÇÃO DOS VANTS ESCRAVOS GUIA- DOS PELO VANT MESTRE FIGURA 23 (D)	65
	ANEXO K ROTA REALIZADA PELO VANT FIGURA 25	67
	ANEXO L GRÁFICOS DA POSIÇÃO DO VANT FIGURA 26 (A)	69
	ANEXO M GRÁFICOS DA POSIÇÃO DO VANT FIGURA 26 (B)	71
	ANEXO N GRÁFICOS DA POSIÇÃO DO VANT ESCRAVO EM RELA- ÇÃO AO MESTRE FIGURA 28 (A)	73
	ANEXO O GRÁFICOS DA POSIÇÃO DO VANT ESCRAVO EM RELA- ÇÃO AO MESTRE FIGURA 28 (B)	75
	ANEXO P GRÁFICOS DA POSIÇÃO DO VANT ESCRAVO EM RELA- ÇÃO AO MESTRE FIGURA 28 (C)	77

1 INTRODUÇÃO

1.1 Motivação

O uso do primeiro veículo aéreo não tripulado (VANT) foi registrado em 1849, há mais de um século (PRISACARIU, 2017). Apesar de antiga, essa tecnologia só se tornou viável nos anos recentes, com a miniaturização de circuitos digitais. Atualmente, com a disponibilidade de dispositivos eletrônicos miniaturizados, diminuiu-se o custo deste tipo de aeronaves e tornou possível a fabricação de VANTs de pequeno porte, o que ocasionou um aumento expressivo na utilização dos mesmos. Aplicações semi-autônomas com estes veículos apresentam grandes desafios técnicos, como desvio de colisão, planejamento de rotas, posicionamento tridimensional preciso e sistemas de controle robustos (IDRIES *et al.*, 2015).

Embarcando os mais diversos sensores nos VANTs, tais como câmeras, sensores de proximidade, sensores térmicos, a aplicabilidade desses pequenos veículos voadores vem aumentando nas mais diversas áreas e tem chamado a atenção da comunidade científica, que procura por novas aplicações. A utilização de múltiplos VANTs de forma colaborativa para realizar uma tarefa oferece algumas vantagens, como agilizar a realização da tarefa, maior área de cobertura e uma possível solução para o problema que é a baixa autonomia de tempo de voo de veículos multirrotores. A literatura apresenta trabalhos na área da agricultura (LEE *et al.*, 2022), inspeção de estações de energia (LU *et al.*, 2022), mapeamento tridimensional de áreas externas (RUI *et al.*, 2022), entre outras.

A manutenção de grande infraestruturas e equipamentos pesados requer constante avaliação para a prevenção de falhas e complicações. Essas examinações normalmente consistem em verificar visualmente pontos chave que podem comprometer a integridade do sistema. Em alguns cenários, porém, esses pontos são de difícil acesso ou insalubres. Mesmo com a utilização de equipamentos especializados, treinamentos e outros cuidados, o bem estar dos funcionários não é garantido. A segurança no trabalho, que vem sendo cada vez mais exigida, pode se beneficiar com o uso de VANTs. Esses, equipados com câmeras, são capazes de obter as imagens necessárias para auxiliar essas avaliações visuais. Diversos tipos de sensores podem ser adicionados para coletar dados adicionais, refinando a precisão de diagnósticos. A mobilidade dos VANTs também é vantajosa em locais de grande altitude, eliminando dificuldades de locomoção e riscos decorrentes. Essa robotização de inspeções é altamente eficaz em situações como torres de distribuição de energia (BERGER, 2020) ou de maçaricos para flaring em petrolíferas (MARINHO *et al.*, 2012).

Um posicionamento preciso é vital para garantir a segurança da operação de multi-VANTs, para garantir o correto seguimento das rotas. O uso de GPS para a navegação é conveniente, porém o uso desta tecnologia exige o recebimento de níveis de sinal apropriados para que opere satisfatoriamente. Em alguns casos, o sinal estável de GPS é difícil de se obter o que inviabiliza a utilização desta técnica (ZIMMERMANN *et al.*, 2017). A diminuição do nível

de sinal além de um nível mínimo, afeta o comportamento do VANT e sua locomoção pode ser imprevisível. Nesses cenários, denominados de ambientes hostis, é preciso utilizar outra forma de navegação para substituir ou auxiliar o GPS.

Uma alternativa à utilização do GPS são técnicas de posicionamento visual. Abordagens comuns fazem o uso de câmeras RGB monocular, binocular e omnidirecional e câmeras estereoscópicas (AL-KAFF *et al.*, 2018). No entanto, a utilização de técnicas de visão computacional para calcular as coordenadas do VANT são computacionalmente custosas.

Uma alternativa para diminuir o custo computacional é criar imagens de marcas para proverem informações específicas da estimação correta da posição do veículo. Etiquetas de realidade aumentada, propostas originalmente para aplicações de realidade aumentada, têm sido utilizadas para esse tipo de aplicação. Exemplos dessas ferramentas são AprilTags (OLSON, 2018) e AR Track Alvar (FINLAND, 2017).

1.2 Objetivos

1.2.1 Objetivos Gerais

Este trabalho tem como objetivo propor uma arquitetura de posicionamento relativo entre múltiplos VANTs, que operam de forma cooperativa. O arranjo será composto por um VANT principal equipado com uma câmera convencional e um ou mais VANTs secundários que possuem etiquetas de realidade aumentada fixadas na parte superior da aeronave. As etiquetas permitem rápida identificação dos veículos secundários e provêm as informações necessárias para o cálculo de suas posições relativas.

1.2.2 Objetivos Específicos

- Elaboração de um ambiente simulado.
- Desenvolvimento de um sistema de posicionamento relativo e cooperativo usando múltiplos VANTs.
- Controle de múltiplos VANTs através do sistema de posicionamento desenvolvido.
- Realizar testes para a validação do trabalho.

2 REVISÃO DE LITERATURA

O processo de visão computacional usado para o cálculo da posição relativa entre dois ou mais veículos semi-autônomos é uma tarefa complexa que comumente requer muita capacidade do hardware. A utilização de etiquetas impressas para fornecer informações para o algoritmo de visão é uma boa alternativa para diminuir a carga computacional. A literatura apresenta algumas pesquisas relacionadas ao uso destes tipos de etiquetas para fornecer informações de posição para controlar o voo de VANTs.

O artigo (HINAS; ROBERTS; GONZALEZ, 2017) propõe uma solução de posicionamento baseado em visão. O trabalho propõe um algoritmo de controle de posicionamento para seguir um alvo em solo representado por uma etiqueta. O trabalho apresenta uma confiança de 92% em experimentos no mundo real, demonstrando a viabilidade deste tipo de abordagem para o posicionamento de VANTs em ambientes exteriores.

Uma proposta de localização em ambientes internos utilizando AprilTag é proposto em (KAYHANI *et al.*, 2019). Um conjunto de marcadores são fixados nas paredes da área de teste provendo uma pista que é capturada pela câmera frontal da aeronave. Um algoritmo executado em uma estação base calcula a posição relativa entre o veículo e cada uma das etiquetas visíveis e envia para o algoritmo de controle do VANT. Apesar da pequena área de voo, esse trabalho confirma a confiabilidade de usar etiquetas para obter as informações de posicionamento tridimensional de VANTs.

Outra abordagem semelhante é apresentada no trabalho (WANG; HU, 2017). A proposta utiliza um VANT semi-autônomo controlado por uma estação base que processa as imagens de um AprilTag obtidas pela câmera do veículo. Os experimentos apresentam controle estável da aeronave utilizando as informações de posição processadas provenientes das etiquetas.

O trabalho apresentado em (BOUDJIT; LARBES, 2015) apresenta um VANT com algoritmo seguidor. Neste trabalho o veículo captura imagens de uma etiqueta impressa segurada em frente ao VANT por uma pessoa. Essas imagens são processadas em uma estação base e fornecem a posição relativa entre o VANT e a etiqueta, permitindo à aeronave a seguir a etiqueta usando um sistema de controle PID.

O uso de etiquetas para prover a posição relativa entre veículos semi-autônomos cooperativos é uma possibilidade para a utilização em aplicações em robótica cooperativa. Trabalhos presentes na literatura propõem o uso dessa ferramenta para pousos autônomos e navegação cooperativa entre robôs de solo e VANTs na maioria dos casos.

O artigo (MUSKARDIN *et al.*, 2016) propõe um sistema de navegação de uso em ambientes externos onde o VANT segue um automóvel em movimento e executa um pouso autônomo. Um conjunto de etiquetas visuais estão fixadas ao carro, permitem o correto posicionamento para o pouso mesmo quando apenas se tem as informações visuais de posicionamento provenientes das etiquetas.

Um sistema de navegação e geração de rotas para um veículo de solo baseado na visão obtida por um VANT de forma cooperativa é proposto em (HOOD *et al.*, 2017). Como uma etiqueta fixada na parte inferior do VANT virada para baixo e uma câmera instalada no topo do robô de solo virada para cima, a aeronave se posiciona em cima do veículo terrestre e o segue. Este sistema de seguimento e posicionamento é baseado apenas nas informações de posicionamento provindas da leitura das etiquetas.

O trabalho realizado em (CANTIERI *et al.*, 2020) apresenta uma proposta de arquitetura para inspeção de linhas de alta tensão utilizando um VANT e um veículo de solo com uma etiqueta Ar-Tag posicionada em seu topo. No sistema proposto os veículos se comunicam para informar sua posição para o outro e a informação de posição proveniente da leitura do marcador é utilizada como um sistema de posicionamento redundante para a aeronave em caso do nível do sinal de GPS caia para valores críticos, de modo aumentar a segurança em tarefas de inspeções.

Este breve estudo da literatura fornece a base para a proposta deste trabalho: o uso de um algoritmo de visão computacional baseado em AR-Tags para permitir a navegação cooperativa entre múltiplos VANTs. O sistema proposto engloba, não somente o sistema de posicionamento, mas também o sistema de controle para a operação simultânea de múltiplos VANTs de forma cooperativa.

3 FUNDAMENTAÇÃO TEÓRICA

A fim de familiarizar o leitor com temas discutidos, técnicas e ferramentas citadas ao longo do texto deste trabalho, uma breve fundamentação teórica será apresentada. Esta abordará conceitos relacionados à robótica e computação.

3.1 ROBÓTICA MÓVEL

A robótica é o campo de estudos que tem como objetivo o desenvolvimento de técnicas e ferramentas relacionadas a robôs e aos desafios que o desenvolvimento e utilização dos mesmos apresentam. Uma das definições possíveis para robô é “Um robô é uma máquina autônoma capaz de perceber o ambiente, realizar cálculos para tomar decisões e fazer ações no mundo real” (GUIZZO, 2018).

Nos últimos anos o desenvolvimento desta área está estreitamente ligado à evolução da sociedade e tem se guiado pelas necessidades que surgem na vida cotidiana. A utilização de robôs tem auxiliado na produtividade de diversas áreas, tal como na indústria automobilística, permitindo que a produção industrial supra a demanda do mercado. Outra aplicação mais recente de sistemas robóticos é a realização de tarefas que expõem pessoas a riscos.

Existem inúmeros tipos de robôs por consequência foram criadas classificações para os mesmos. A classe de interesse deste trabalho é a robótica móvel, nesta se encontram os robôs que têm a capacidade de se locomover no ambiente. O modo pelo qual a locomoção é realizada varia com o ambiente e com a tarefa que o robô necessita realizar. A habilidade de se mover abre novas áreas a serem estudadas, tais como problemas de navegação, mapeamento de ambientes, posicionamento, entre outras (GARCIA *et al.*, 2007).

A movimentação de robôs traz novos problemas. O problema primordial é a movimentação em si, o estudo deste problema é chamado de cinemática. Antes de se locomover é preciso se localizar no espaço, para isso é necessário saber a posição e a orientação do robô e o conjunto dessas duas informações foi nomeado de *pose*. A *pose* é um dos principais parâmetros usado para realizar o controle dos sistemas robóticos.

A *pose* do robô sempre se relaciona com um referencial. Pode-se utilizar vários objetos como referencial, como por exemplo uma câmera ou outro sensor, outro robô e o mais comum é utilizar o próprio ambiente como referencial. O ambiente pode ser considerado o planeta inteiro como quando se utiliza sistemas de posicionamento global (GPS) com informações de latitude e longitude, para obter a *pose*. Ou utilizar apenas o ambiente em torno do robô como referência, chamado de sistema de posicionamento local. Nesse caso a *pose* do robô é baseada em informações provenientes de sensores inerciais ou técnicas de posicionamento que se utilizam de outros tipos de sensores que calculam o deslocamento a partir de um ponto inicial, o qual é o ponto onde o sistema robótico foi inicializado.

Existem várias formas de se representar a posição e a orientação de um objeto no espaço. Algumas dessas favorecem as manipulações e cálculos realizados por computadores, por exemplo uma das formas mais utilizadas para se representar posição é através de um vetor de duas ou três dimensões. Os valores contidos no vetor representam a distância em cada eixo do objeto em questão em um plano cartesiano, plano esse que é centrado em um referencial escolhido, como pode-se observar na Equação 1.

$$p_i = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad (1)$$

Para representar a rotação, a maneira mais intuitiva é utilizando ângulos, sejam em graus ou radianos. Esta representação se chama Ângulos de Euler explicada em Siciliano, Khatib e Kröger (2008). Porém a utilização desta representação em sistemas computacionais não é a mais comum. Pois a manipulação das informações envolvem multiplicação de matrizes, que são computacionalmente custosas, e também necessita cálculos de senos e cossenos o que pode gerar problemas de representação uma vez que as casas decimais em um computador são limitadas. Uma solução possível para contornar estes problemas é a utilização de um método de representação chamado de Quaternio. Este modelo foi sugerido por (HAMILTON, 1848). Nesta representação a orientação é descrita como um número complexo, como visto na Equação 2.

$$\alpha = a_0 + a_1i + a_2j + a_3k \quad (2)$$

Onde a_0 , a_1 , a_2 e a_3 são componentes escalares e i , j e k são operadores que satisfazem as seguintes regras,

$$i^2 = j^2 = k^2 = -1, \quad (3a)$$

$$ij = k, jk = i, ki = j, \quad (3b)$$

$$ij = -k, jk = -i, ki = -j \quad (3c)$$

Este modelo de representação diminui o custo computacional da computação da orientação de um objeto, sendo a mais comumente utilizada na área da robótica (SICILIANO; KHATIB; KRÖGER, 2008).

3.2 VEÍCULO AÉREO NÃO TRIPULADO

Os veículos aéreos não tripulados (VANTs), são robôs aéreos voadores que não carregam um operador humano junto a ele. Essas aeronaves não tripuladas podem assumir as mais diversas formas dependendo do propósito a qual ela será utilizada. Dentro desse vasto

grupo chamado de VANT ainda existem inúmeros subgrupos onde esses veículos são agrupados levando em conta suas características, tais como modo de propulsão, número de propulsores, modo de voo, entre outras (HASSANALIAN; ABDELKEFI, 2017). Na Figura 1 ilustra-se um VANT de asa fixa.

Figura 1 – Exemplo de VANT de asa fixa.



Fonte: (SKULSTAD et al., 2015).

A subcategoria de VANT multirrotores de asa rotativa ilustrado na Figura 2. É caracterizada por utilizar múltiplos propulsores rotativos. São veículos ágeis, que realizam decolagem e pouso na vertical. Apesar da pequena capacidade de carga são amplamente utilizados em diversas atividades, desde lazer e *hobby* até aplicações comerciais e industriais (KARDASZ; DOSKOCZ, 2016).

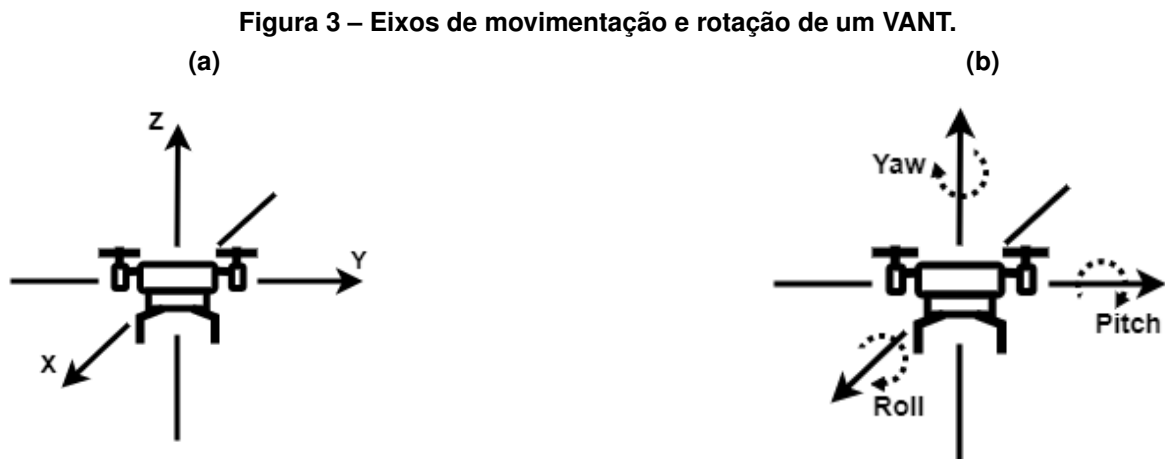
Figura 2 – Exemplo de VANT multirrotor de asa rotativa.



Fonte: Autoria própria.

Uma característica marcante desse tipo de aeronave é o modo como ela se movimenta. Tomando sua estrutura como referência tem-se três eixos possíveis de movimentação e rotação

como ilustra a Figura 3 (a) e (b). Para elaborar o modelo de movimentação da aeronave define-se um plano cartesiano com três eixos centralizado no meio da estrutura do VANT e um outro plano cartesiano centralizado em um ponto que se deseja que seja o referencial para os cálculos da pose dele. Esta pose é representada por meio de um vetor que representa a posição da aeronave em relação ao referencial adotado e um quaternio que indica a rotação do VANT em relação ao referencial.



Fonte: Autoria própria.

Para se movimentar no eixo Z, ou seja na vertical, o VANT varia a velocidade dos motores de modo a variar a força de sustentação desta forma permitindo que ele vá para cima ou para baixo. Para realizar movimentos de rotação em torno de seu próprio eixo, movimento esse comumente chamado de yaw (em tradução livre “guinada”), as velocidades dos motores são alteradas. Mantendo constante a força que sustenta a aeronave no ar e causando um desequilíbrio nos torques que os motores causam na estrutura do VANT fazendo com que surja uma velocidade angular no mesmo. Para realizar os movimentos ao longo dos eixos X e Y, novamente são manipuladas as velocidades dos motores. A alteração de velocidade neste caso mantém em equilíbrio a altitude e a rotação de yaw, mas provoca alterações nas rotações do eixo X e Y, estes eixos de rotação são chamados comumente de roll e pitch respectivamente (em tradução livre “rolagem” e “levantar”). Alterando a inclinação nesses sentidos faz com que surja uma componente horizontal da força que sustenta o VANT no ar produzindo assim uma aceleração que faz com que ele realize movimentos horizontais (AMIR; ABBASS, 2008).

3.3 VISÃO COMPUTACIONAL

Visão computacional é uma área da Inteligência Artificial que permite que dispositivos inteligentes extraiam alguma informação a partir de imagens ou vídeos. É ela a responsável por simular a habilidade da mente humana de reconhecer objetos, estimar distâncias e profundidade. Para isso é necessário uma grande quantidade de dados e poder de processamento.

Análises de dados são feitas e refeitas constantemente até que as distinções sejam reconhecidas e, conseqüentemente, o objeto seja identificado (MARENGONI; STRINGHINI, 2009).

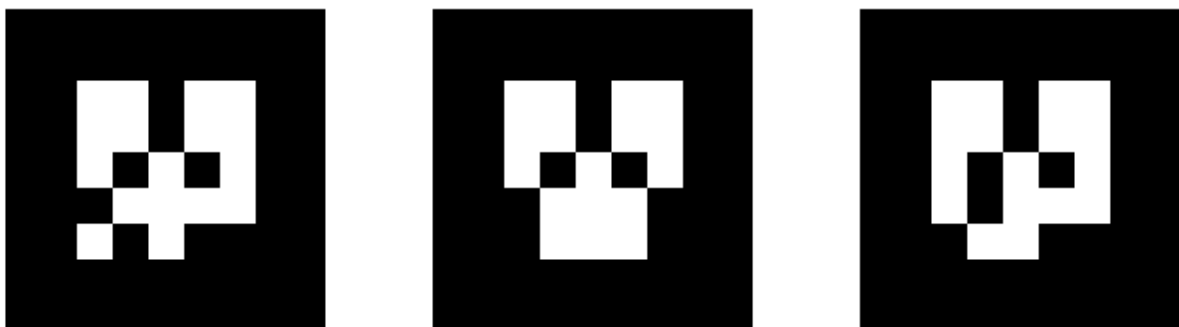
3.4 PROCESSAMENTO DIGITAL DE IMAGENS

Quando uma imagem é analisada na visão computacional, ela passa por uma série de operações para extrair informação. Essas operações variam de acordo com o objetivo, podendo ser algo simples como ajuste de brilho e contraste, ou processos mais complexos como realçador de contorno (MARENGONI; STRINGHINI, 2009).

3.5 RASTREAMENTO BASEADO EM MARCADORES

Diversos métodos de rastreamento foram desenvolvidos ao longo do tempo nas mais diversas áreas. O termo rastreamento diz respeito ao cálculo da posição, tanto localização como orientação, da câmera em relação à um referencial. Uma das técnicas é o rastreamento visual, o sistema obtém a posição da câmera se baseando na observação do ambiente por meio da captura de imagens. Esse processo pode se tornar desafiador se realizado em um ambiente desconhecido. Uma solução possível é a adição de marcadores, ilustrados na Figura 4, no cenário e usar técnicas de visão computacional para detectá-los. Uma vez detectada a etiqueta, por meio de processamento de imagem, pode-se obter informações da posição dela em relação à câmera. Esta abordagem é chamada de rastreamento baseado em marcadores (SILTANEN, 2012).

Figura 4 – Etiquetas utilizadas pelo algoritmo Alvar.



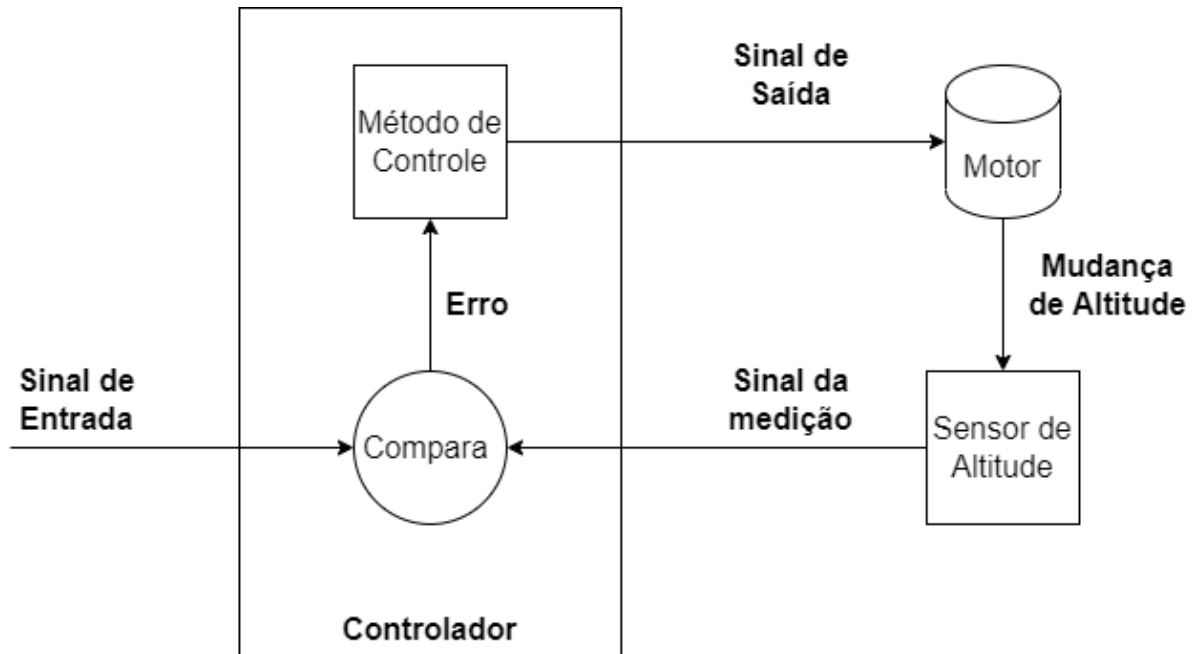
Fonte: http://wiki.ros.org/ar_track_alvar.

3.6 CONTROLADORES PID

Controladores são dispositivos intermediários de um processo, onde deseja-se minimizar a diferença entre sinais de entrada e de saída. Para isso, um sinal é gerado pela medição de um sensor e comparado com o sinal de entrada. Essa diferença entre sinais determina um

signal de saída apropriado para gerar uma ação de correção no elemento controlado (KNOSPE, 2006). A Figura 5 apresenta um exemplo de um diagrama de controle para um VANT.

Figura 5 – Diagrama de um controlador de altitude usado em VANTS.



Fonte: Autoria própria.

Existem vários métodos para determinar um sinal de saída adequado. Um destes é o PID, que utiliza três parâmetros: proporcional, integral e derivativo. Neste, o sinal de saída é a combinação do sinal gerado por cada parâmetro. Estes três componentes atuam independentemente e contribuem em finalidades diferentes.

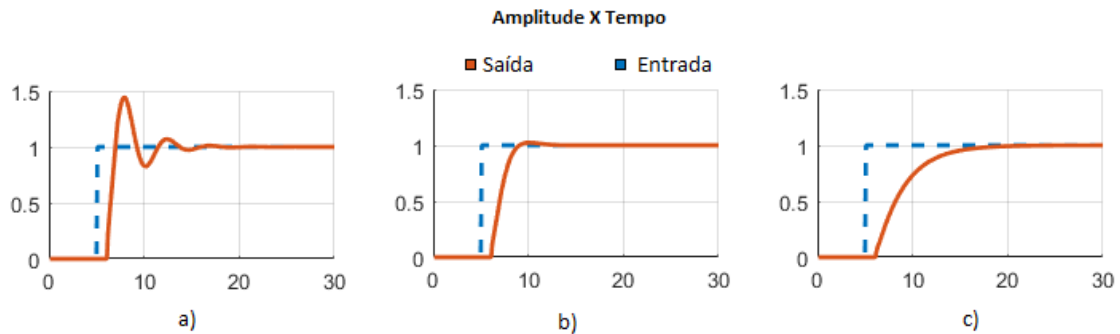
3.6.1 Controlador Proporcional

De todos os controladores, o proporcional é o mais simples. O sinal de saída é a diferença entre sinal medido e o sinal de entrada (erro) multiplicado por uma constante. Esta, denominada K_P , possui um valor arbitrário que deve ser escolhido de acordo com o comportamento desejado da saída. É possível observar o tempo de resposta do sistema observando a Figura 6. Quanto menor o valor de K_P , mais tempo levará para que o sistema se ajuste ao valor desejado. E, da mesma forma, quanto maior o valor de K_P mais rápido será este ajuste. É preciso ter em mente que valores muito altos de K_P podem gerar oscilações na resposta, o que é indesejável em um controlador de altura.

3.6.2 Controlador Integral

O controlador integral serve como um suplemento ao controlador proporcional. Independente do valor escolhido da constante, o controlador proporcional por si só nunca consegue

Figura 6 – Gráficos representando o comportamento do sinal de resposta ao longo do tempo para valores diferentes de K_P . a) K_P alto, resposta rápida e comportamento oscilatório. b) K_P ideal. c) K_P baixo, resposta lenta.



Fonte: Autoria própria.

chegar exatamente ao valor desejado e, por isso, sempre existirá uma defasagem proporcional a K_P . Para obter um resultado mais preciso, o controlador integral usa a somatória dos erros para multiplicar pela constante K_I . O resultado é um controlador que corrige erros com longa duração, como a defasagem do controlador proporcional. Assim como K_P , o valor de K_I deve ser escolhido dependendo do problema, para que não haja instabilidade.

3.6.3 Controlador Derivativo

Para grande parte dos problemas um controlador PI é suficiente, mas nem tanto para sistemas cujo comportamento é afetado por fatores externos. O controlador derivativo é usado em conjunto nesse tipo de aplicação para estabilizar variações no sinal de saída devido à esses fatores. Neste, a variação do erro é multiplicada por K_D para gerar o sinal de saída. Valores maiores de K_D geram uma resposta mais intensa à mudanças bruscas no sinal medido. Nota-se que por ser muito sensível à variações, o controlador derivativo geralmente é acompanhado de um filtro para amenizar o ruído na leitura do sinal.

3.6.4 Algoritmo Ziegler–Nichols

No controlador PID, as três constantes: K_P , K_I e K_D , devem ser escolhidas de acordo com os requisitos do sistema. Existem diversos métodos de ajuste para determinar esses valores. Neste projeto utilizou-se o algoritmo Ziegler–Nichols (ZIEGLER; NICHOLS, 1942). Nele, consegue-se calcular os parâmetros a partir de medições, tornando-o prático. Este é um processo que procura reduzir a oscilação da resposta e manter uma velocidade de resposta relativamente alta. Começa-se zerando os valores de K_I , K_D e K_P , aumentando K_P até que o sinal de resposta tenha um comportamento oscilatório e uma amplitude consistente. Quando isso

ocorrer denomina-se K_U o valor atual de K_P e, T_U o tempo do período de oscilação. A partir dessas medições, as três constantes são calculadas como mostra as equações 4, 5 e 6.

$$K_P = 0.60K_U \quad (4)$$

$$K_I = \frac{1.2K_U}{T_U} \quad (5)$$

$$K_D = \frac{3K_U T_U}{40} \quad (6)$$

3.7 SIMULAÇÃO ROBÓTICA

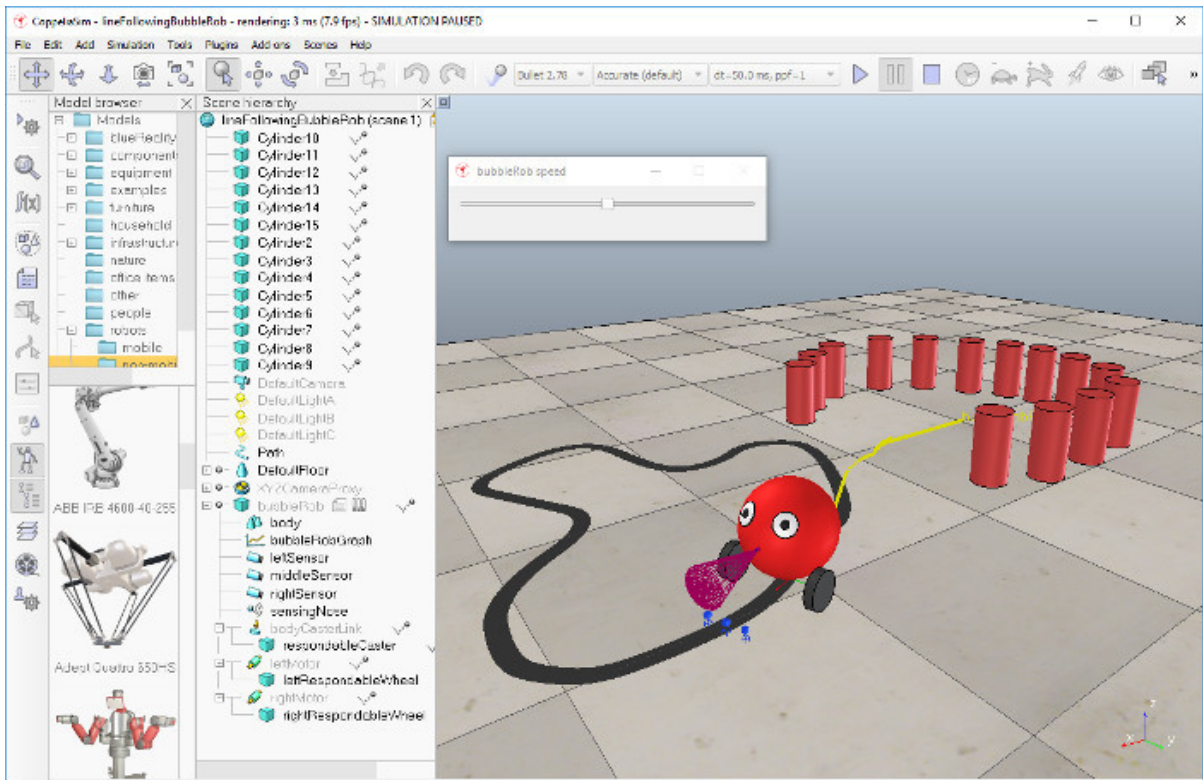
Simulação é realizar um experimento com um modelo. Um modelo é a representação de um sistema com alguma finalidade, e em geral é uma simplificação do objeto original mas que contenha um nível de detalhamento necessário para satisfazer o objetivo do estudo. Este método é uma ferramenta importante em ocasiões onde o estudo tem limitações para ser realizado tais como o projeto é muito caro para se realizar, ou sua implementação é muito demorada ou a condução dos experimentos não é segura (WHITE; INGALLS, 2018).

Os estudos na área da robótica assim como suas aplicações tem evoluído e crescido rapidamente nos últimos anos. E observa-se que nos próximos anos essa tendência tende a aumentar cada vez mais. Para tornar as pesquisas e desenvolvimento menos dispendioso, cada vez mais ferramentas de simulação de robótica vem surgindo. Além de baratear os estudos, a utilização de simuladores agiliza o desenvolvimento e facilita a realização de teste. E cada vez mais os desenvolvedores desses programas tentam torná-los mais próximos da realidade quanto a computação atual permite, para que os sistemas robóticos desenvolvidos em simulação sejam mais compatíveis com o mundo real (KRUPKE *et al.*, 2012).

3.7.1 CoppeliaSim

O simulador de robótica CoppeliaSim é um ambiente de desenvolvimento de sistemas robóticos e algoritmos e atualmente um dos mais utilizados. Apesar de ser um *software* comercial o uso para fins educacionais e de pesquisa é gratuito. Esse simulador utiliza varias bibliotecas de simulação de física e utiliza algoritmos de cinemática e dinâmica para realizar a simulação. Oferece vários recursos que facilitam o desenvolvimento dos projetos e o torna muito versátil. Pode ser utilizado sozinho, já que permite o uso de scripts em Python e LUA embarcados nos objetos simulados, ou em conjunto com outras ferramentas (ROHMER; SINGH; FREESE, 2013) (ROBOTICS, 2019). A Figura 7 ilustra a interface de desenvolvimento do CoppeliaSim.

Figura 7 – Exemplo de ambiente desenvolvido no CoppeliaSim.



Fonte: (ROBOTICS, 2019).

3.8 ROBOT OPERATING SYSTEM

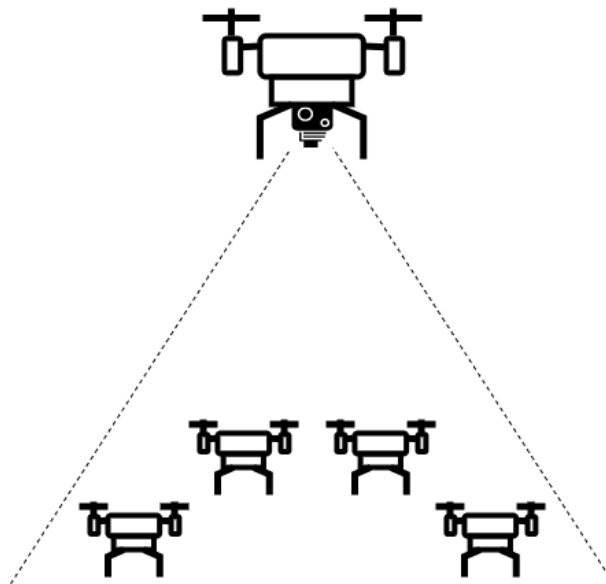
Robot operating system (ROS) é uma plataforma open-source que consiste em um framework que fornece um conjunto de ferramentas e bibliotecas permitindo a abstração do controle do hardware no desenvolvimento de sistemas robóticos complexos. (PYO *et al.*, 2017)

Com uma comunidade ativa, várias ferramentas e bibliotecas são desenvolvidas e compartilhadas, tornando mais rápido e simples o desenvolvimento de sistemas uma vez que vários módulos podem ser reutilizados em diversos projetos diferentes. Este processo de reutilização de módulos também ajuda o desenvolvimento das ferramentas, uma vez que estas são revisadas e melhoradas ao longo do tempo à medida que são utilizadas em diversos projetos.

4 ARQUITETURA DO SISTEMA

A arquitetura do modelo de posicionamento baseado em visão, desenvolvido neste trabalho, é composto por um VANT mestre que captura imagens de outros VANTs que estão abaixo dele, os VANTs escravos, como ilustrado na Figura 8. Nos tópicos a seguir serão apresentadas mais detalhadamente as arquiteturas e componentes utilizados tanto na implementação do sistema em ambiente simulado quanto no mundo real.

Figura 8 – Arquitetura simplificada do sistema.



Fonte: Autoria própria.

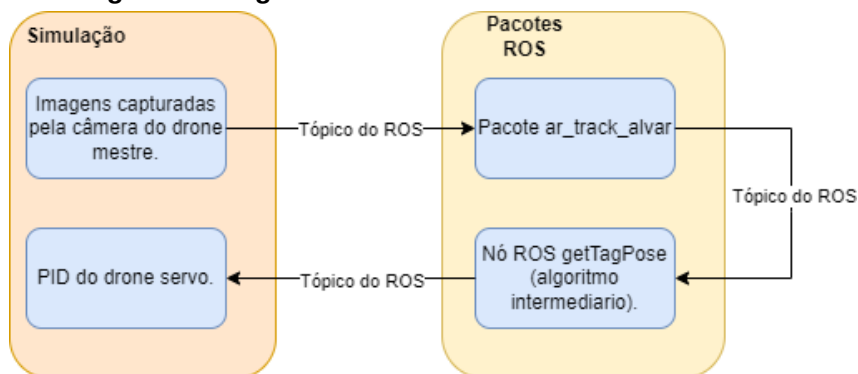
4.1 SIMULAÇÃO

A simulação foi desenvolvida utilizando o ambiente de desenvolvimento CoppeliaSim 4.3.0, tendo o sistema operacional base o Windows 11 e utilizando o Ubuntu 20.04 LTS por meio de WSL 2 (*Windows subsystem for Linux*) em um *notebook* com a seguinte configuração: Intel i7, 16GB de memória RAM e placa gráfica dedicada NVIDIA GeForce RTX 3070.

O sistema de posicionamento baseado em visão desenvolvido utiliza um VANT master com uma câmera posicionada para baixo para aquisição de imagens das etiquetas fixadas nos VANTs escravos posicionados abaixo dele. Como a posição relativa dos VANTs é obtida pela imagem da câmera, é importante que ela esteja o mais estável possível, pois uma pequena variação da câmera pode gerar uma medida incorreta. Para isto optou-se por fixar a posição e angulação da câmera em relação ao VANT, simulando um sistema físico de estabilização de imagem. Dessa forma, qualquer tipo de ruído é eliminado, tornando a câmera mais estável e evitando o aumento do erro de medida da posição dos VANTs escravos.

As imagens captadas no VANT simulado são processadas pelo algoritmo Alvar contido no pacote do ROS *ar_track_alvar* e as informações de posicionamento são publicadas em um tópico ROS. Estas informações são repassadas para o sistema de controle dos VANTs escravos para que eles mantenham a posição relativa ao VANT mestre ou realizam a movimentação exigida pela tarefa designada à eles. Por uma incompatibilidade dos tipos de mensagem publicada pelo tópico *ar_track_alvar* e os tipos de mensagem que o simulador aceita, foi necessário o desenvolvimento de um algoritmo intermediário para que a comunicação entre o algoritmo Alvar e o simulador acontecesse de forma correta. Este algoritmo intermediário foi implementado em C++ como um nó de ROS. Ele lê o tópico */ar_pose_marker* que informa a posição da etiqueta, fornecido pelo pacote *ar_track_alvar* que é do tipo “*AlvarMarkers Message*”. Extrai as informações de posição e orientação e republica em outros dois tópicos sendo eles um contendo as informações de posição no tipo de mensagem “*Point Message*” e o outro contendo o yaw no tipo de mensagem “*Float32 Message*”. Ambas as mensagens republicadas são estruturas de comunicação presentes no ROS. Para facilitar entendimento a Figura 9 demonstra o sistema descrito.

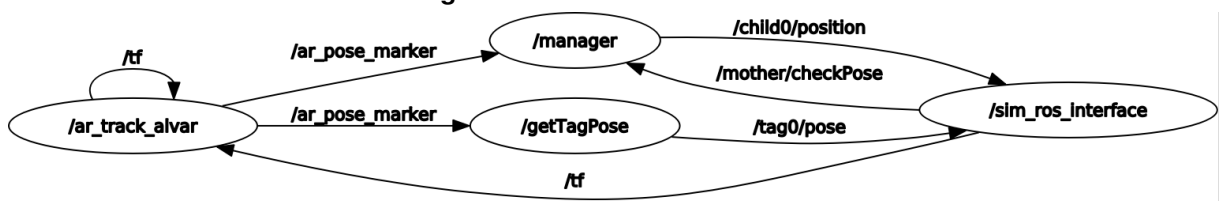
Figura 9 – Diagrama do sistema simulado com um VANT.



Fonte: Autoria própria.

A integração realizada pelo ROS pode ser visualizada na Figura 10 que apresenta os nós de ROS do sistema e suas ligações. Os circulos são os nós do ROS e as setas representam os tópicos ROS que fazem a comunicação entre os componentes do sistema. O nó *sim_ros_interface* é responsável pela interface entre o simulador e o ROS, o nó *manager* é o que envia as posições que os VANTs devem manter.

Figura 10 – Grafo dos nós ROS.

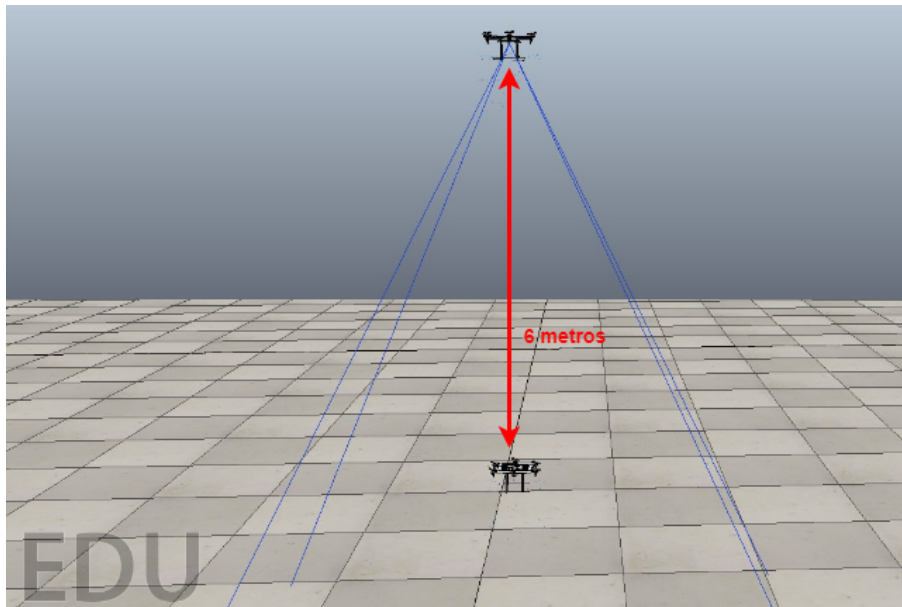


Fonte: Autoria própria.

Como no ambiente simulado tem-se uma simplificação do mundo real, não há tantos fatores que interferem no bom funcionamento de componentes do sistema, tal como a variação da iluminação no mundo real que interfere na leitura da câmera das etiquetas. Isso possibilitou posicionar os VANTs mais afastados um do outro. Como exemplificado pela Figura 11, no primeiro momento dois VANTs foram colocados alinhados a uma distância de 6 metros um do outro. Esta foi a primeira cena elaborada para serem realizadas medidas dos erros de medida referentes às etiquetas e testes do sistema de controle.

A movimentação dos VANTs foi realizada a partir do envio de pontos objetivos para eles. O sistema de controle de posicionamento do VANT mestre tem como entrada a posição global obtida do simulador e o controle dos VANTs escravos é alimentado pelas informações de posições provenientes das leituras das etiquetas.

Figura 11 – Primeira cena desenvolvida no CoppeliaSim.



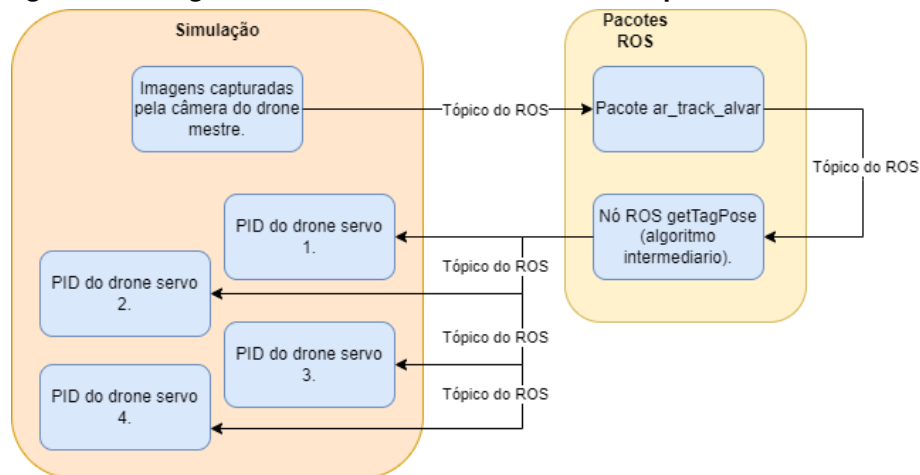
Fonte: Autoria própria.

Em um segundo momento foram adicionados mais três VANTs escravos, totalizando quatro VANTs escravos, como ilustra a Figura 13. As etiquetas utilizadas pelo algoritmo Alvar podem ser geradas em padrões diferentes e são identificadas por um identificador único, que permite a identificação de cada um dos VANTs escravos pois eles possuem etiquetas diferentes.

O funcionamento do sistema simulado com múltiplos VANTs escravos é análogo ao funcionamento do sistema com apenas um escravo. Como pode ser visto na Figura 12 a diferença é que o algoritmo intermediário envia informações para múltiplos algoritmos de PID para as múltiplas aeronaves. Como essa alteração é apenas no ambiente da simulação ela fica encapsulada no nó do ROS da simulação. Assim a estrutura dos nó ROS se mantém como representado na Figura 10

As informações de posicionamento dos marcadores, fornecidas pelo nó de ROS `/ar_track_alvar`, são organizadas e separadas em função do identificador da etiqueta. O nó ROS

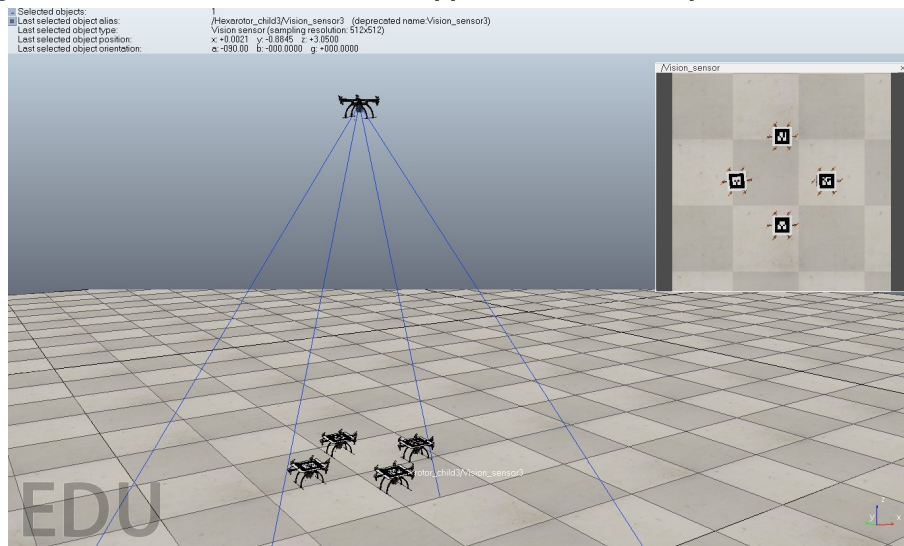
Figura 12 – Diagrama do sistema simulado com múltiplos VANTs escravos.



Fonte: Autoria própria.

getTagPose, responsável pelo intermédio da comunicação entre o algoritmo Alvar e o CoppeliaSim, faz a identificação das etiquetas do sistema e a separação dos dados de cada um dos marcadores. Além disso, cria tópicos de ROS específicos para cada um dos VANTs escravos, fazendo com que cada um dos sistemas de controle das aeronaves recebam apenas as informações pertinentes à ela.

Figura 13 – Cena desenvolvida no CoppeliaSim com múltiplos VANTs escravos.



Fonte: Autoria própria.

4.2 IMPLEMENTAÇÃO NO MUNDO REAL

Os testes reais começaram utilizando uma câmera, *Webcam* HD Logitech C920-1080p, fixa em uma posição e posicionada para baixo. Foram tomadas medidas em cinco posições diferentes utilizando etiquetas de três tamanhos diferentes, 5 cm, 7 cm e 10cm. Foram feitas

cem medidas em cada um dos pontos que foram utilizadas para calcular o erro de leitura do posicionamento utilizando o algoritmo Alvar no mundo real.

Figura 14 – Experimento realizado com apenas um VANT real.



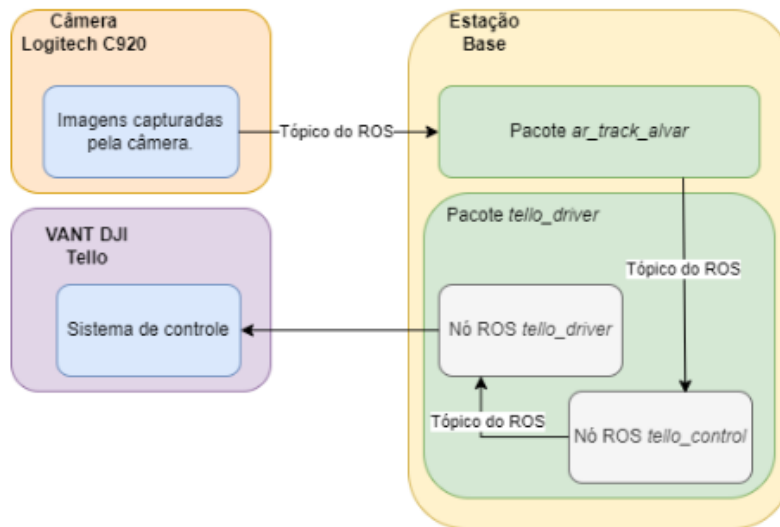
Fonte: Autoria própria.

O próximo passo foi o teste com um VANT real, do modelo DJI Tello mostrado na Figura 17b. O controle de posicionamento foi feito através do uso de uma etiqueta de 7 cm posicionada na parte de cima do VANT e a mesma câmera fixada como visto na Figura 14. Este experimento foi feito utilizando um notebook Intel i7, 16GB de memória RAM e placa gráfica integrada Intel HD Graphics 520 (Skylake GT2) como estação base. O VANT cria uma rede *Wi-Fi* 2.4GHz à qual a estação base se conecta, estabelecendo a comunicação entre eles. A interface de comunicação do sistema ocorre por intermédio de nós e tópicos do ROS, que enviam mensagens da câmera para a estação base e da estação base para o VANT. Um algoritmo de controle foi desenvolvido e é executado no computador base e envia comandos para o VANT por intermédio de um pacote ROS, *tello_drive* disponibilizado por Xu (2018), responsável por interpretar as mensagens do sistema de controle e enviar comandos à aeronave.

A Figura 15 mostra a arquitetura para este sistema. A câmera faz a captura das imagens, envia para a estação base através de um tópico do ROS. Na estação base as imagens são processadas pelo algoritmo Alvar que extrai as informações de posição, passa essas informações para o algoritmo do sistema de controle e esse manda um comando de velocidade do tipo *cmd_vel* para o VANT.

A comunicação entre os nós do ROS é apresentada na Figura 16. Pode-se observar o fluxo de funcionamento do sistema descrito acima. O nó *usb_cam* é responsável pela aquisição das imagens. O nó *ar_track_alvar* processa a imagem para a identificação da Tag. O nó *tello_ID1/pid_control* contém o algoritmo de controle. O nó *tello_ID1_drive_node* converte as

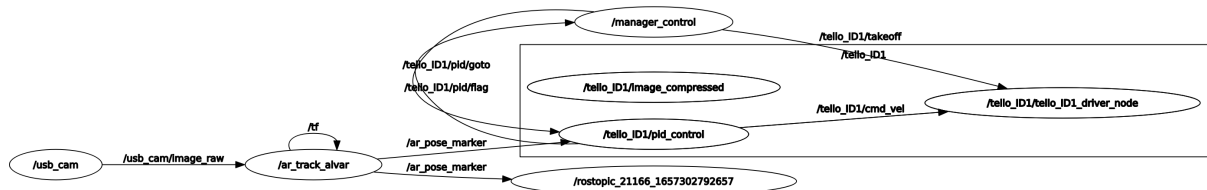
Figura 15 – Diagrama do sistema elaborado para teste com um VANT.



Fonte: Autoria própria.

mensagens e as envia para o VANT. O nó *manager_control* é uma interface contendo algoritmos desenvolvidos para gerenciar as rotinas de testes.

Figura 16 – Grafo dos nós do ROS.



Fonte: Autoria própria.

O último experimento realizado foi a utilização de dois VANTs reais. Um Parrot Bebop como o VANT mestre e um DJI Tello como VANT escravos, ambos ilustrados na Figura 17. Ambas as aeronaves funcionam com a arquitetura de rede *hot-spot Wi-Fi*, portanto foram utilizados dois notebooks como estações bases. Ambos têm a configuração Intel i7, 16GB de memória RAM e placa gráfica integrada Intel HD Graphics 520 (Skylake GT2). O computador conectado ao Bebop utilizava o sistema operacional Ubuntu 16.04 LTS e o ROS Kinect, já a estação base conectada ao Tello utilizava o sistema operacional Ubuntu 20.04 LTS e o Ros Noetic.

A comunicação entre o VANT mestre e sua estação base é feita através do pacote ROS Bebop Autonomy, (MONAJJEMI, 2015), e a comunicação com o Tello se mantém pelo pacote *tello_drive*. Com a câmera do Bebop posicionada para baixo, ela captura imagens da etiqueta posicionada em cima do VANT escravo, o Tello. Essas imagens são enviadas via tópico de ROS para a estação base respectiva onde está sendo executado o algoritmo Alvar por meio do pacote ROS *ar_track_alvar* que obtém as informações de posição do VANT escravo. Essas informações são repassadas para a outra estação base onde está sendo executado o algoritmo de controle do Tello e os comandos de velocidade gerados pelo sistema de controle são enviados via tópico

Figura 17 – VANTs utilizados nos experimentos reais.

(a) Parrot Bebop



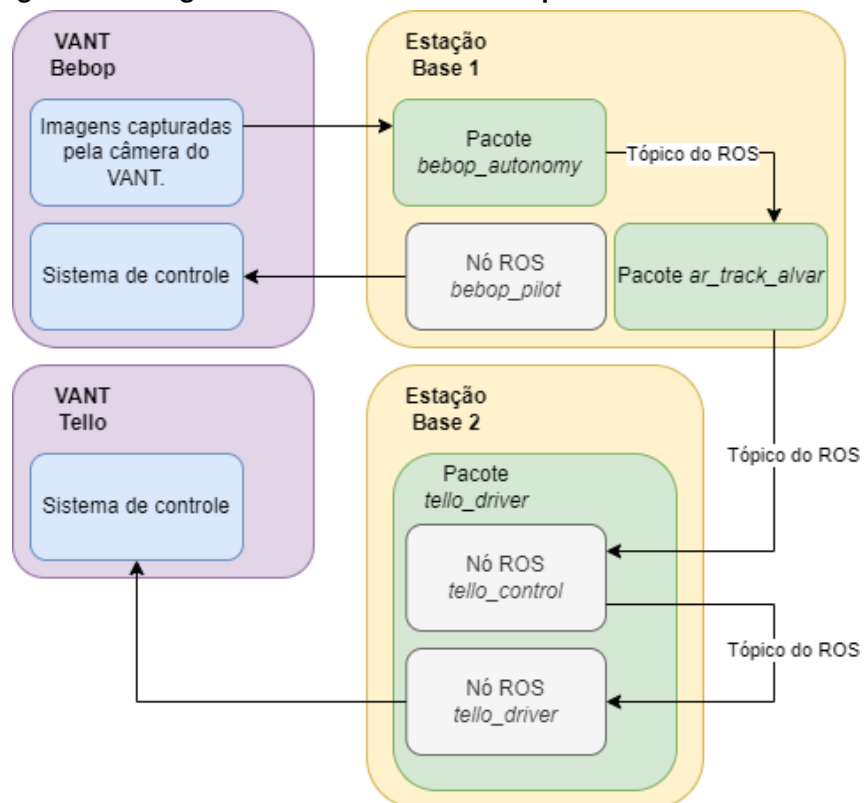
(b) DJI Tello



Fonte: a) Popper (2015) b) Autoria própria.

de ROS para o pacote *tello_drive* que os envia para o VANT escravos. O funcionamento do sistema explicado acima pode ser visualizado na Figura ??.

Figura 18 – Diagrama do sistema elaborado para teste com dois VANTs.



Fonte: Autoria própria.

5 EXPERIMENTOS E RESULTADOS

Os testes executados tiveram como objetivos principais quantificar o erro na leitura da posição pelo algoritmo Alvar e validar o sistema de posicionamento baseado em visão proposto. Os experimentos foram conduzidos seguindo a seguinte ordem:

- Testes em ambiente simulado.
 - Teste com um VANT escravo executando rotas pré definidas com o VANT mestre mantendo sua posição fixa.
 - Teste com um VANT escravo mantendo sua posição relativa ao VANT mestre, enquanto este executa rotas pré definidas.
 - Teste com múltiplos VANTs escravos executando rotas pré definidas com o VANT mestre mantendo sua posição fixa.
- Testes em ambiente real.
 - Coleta de medidas com etiquetas de três tamanhos diferentes em cinco posições diferentes e a câmera fixa.
 - Teste com um VANT escravo realizando uma rota pré definida e a câmera fixa.
 - Teste com um VANT mestre e um escravo que mantém a posição relativa ao mestre.

5.1 AMBIENTE SIMULADO

5.1.1 Experimento com um VANT escravo

O primeiro experimento realizado em ambiente simulado com apenas um VANT mestre que mantém uma posição fixa a 9 m de altura e um VANT escravo que realizou três rotas diferentes mantendo sua altitude em 3 m. As rotas realizadas pela aeronave serva são ilustradas pelos gráficos apresentados nas Figura 19 e Figura 20. A partir dos pontos definidos sabe-se a rota que o VANT deveria executar. As informações de posição da leitura da etiqueta e da posição real do VANT escravo, retiradas do CoppeliaSim, foram salvas em arquivo csv.

O objetivo desse experimento é a validação do sistema de controle de posicionamento na movimentação de apenas um VANT escravo, e aferição do erro de posicionamento na aquisição de informações do algoritmo Alvar. Para calcular os erros de medidas de posicionamento dos marcadores foram utilizados os valores do posicionamento global do simulador como parâmetro de comparação. Os valores obtidos são apresentados na Tabela 1.

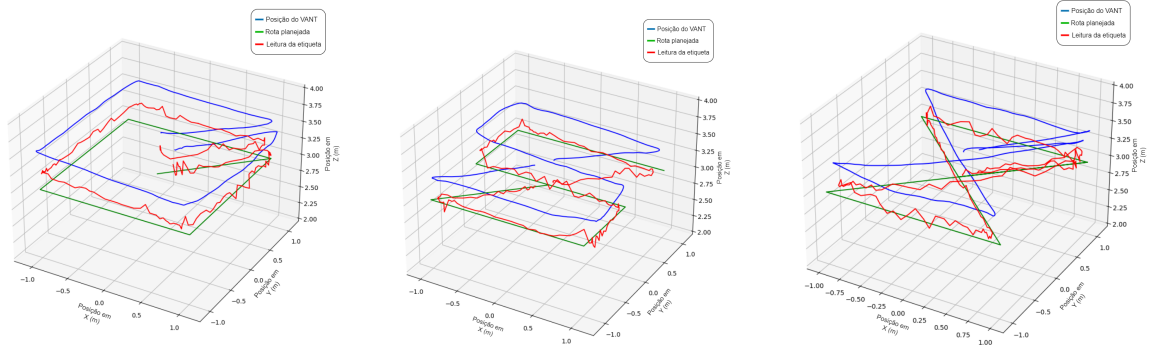
Na Figura 19 pode-se observar que a rota representada pela linha azul, que é a posição real da aeronave obtida pela posição global obtida do simulador, está constantemente acima

Figura 19 – Gráficos da posição do VANT escravo em relação ao VANT mestre. Imagens ampliadas nos Anexos A, B e C

(a) Rota 1.

(b) Rota 2

(c) Rota 3



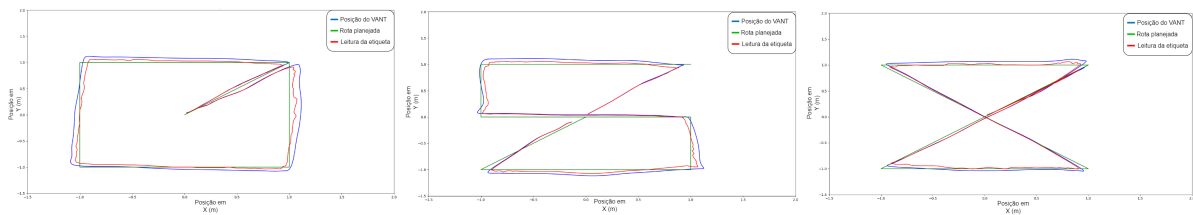
Fonte: Autoria própria.

Figura 20 – Gráficos da posição dos eixos X e Y do VANT escravo em relação ao VANT mestre. Imagens ampliadas nos Anexos D, E e F

(a) Rota 1.

(b) Rota 2

(c) Rota 3



Fonte: Autoria própria.

das demais. Esse fato pode ser também notado nos valores dos erros calculados, observa-se que os maiores erros são encontrados nas medidas do eixo Z. Isso ocorre recorrentemente em todos os experimentos que foram realizados em ambiente simulado. A origem deste erro provavelmente é a distorção que ocorre na imagem da etiqueta quando é inserida em uma superfície no CoppeliaSim.

Também foi testada a capacidade do VANT escravo seguir o VANT mestre, ilustrado na Figura 21. Este funcionamento é uma característica importante, que permite que a aeronave mestre conduza por exemplo as aeronaves servas até a área onde eles realizam uma tarefa ou as conduza até a estação base.

5.1.2 Experimento com múltiplos VANTs escravos

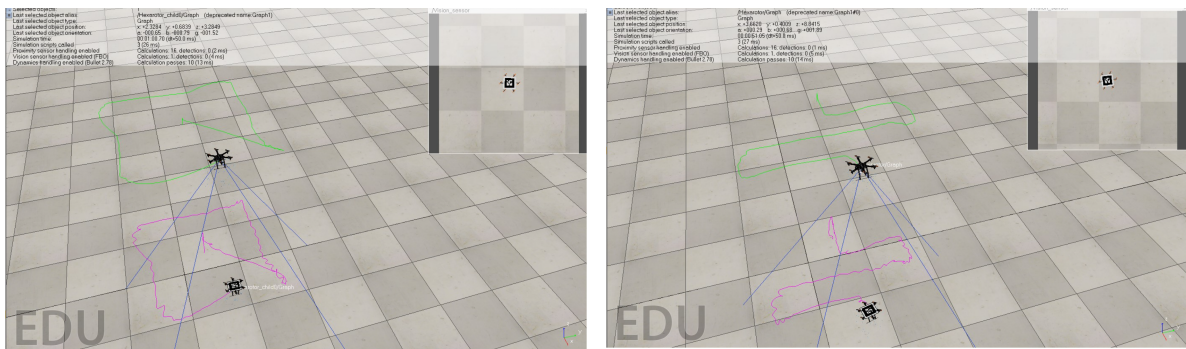
No experimento com múltiplos VANTs escravos sendo guiados pelo VANT mestre na execução de uma rota predefinida, como ilustrado na Figura 22, obteve-se resultados parecidos com os testes já realizados. Dois dos VANTs escravos foram instruídos a subir 50 cm, para a altitude de 3,5 m e realizarem um quadrado de um metro de lado. Os outros dois foram

Tabela 1 – Erros das medidas nas três rotas.

Rota 1			
	Eixo X	Eixo Y	Eixo Z
Erro médio (m)	0,011	0,010	0,0315
Desvio padrão do erro (m)	0,042	0,042	0,041
Erro máximo (m)	0,085	0,089	0,423

Rota 2			
	Eixo X	Eixo Y	Eixo Z
Erro médio (m)	0,012	0,012	0,312
Desvio padrão do erro (m)	0,037	0,042	0,046
Erro máximo (m)	0,089	0,087	0,466

Rota 3			
	Eixo X	Eixo Y	Eixo Z
Erro médio (m)	0,005	0,005	0,037
Desvio padrão do erro (m)	0,039	0,045	0,316
Erro máximo (m)	0,090	0,091	0,457

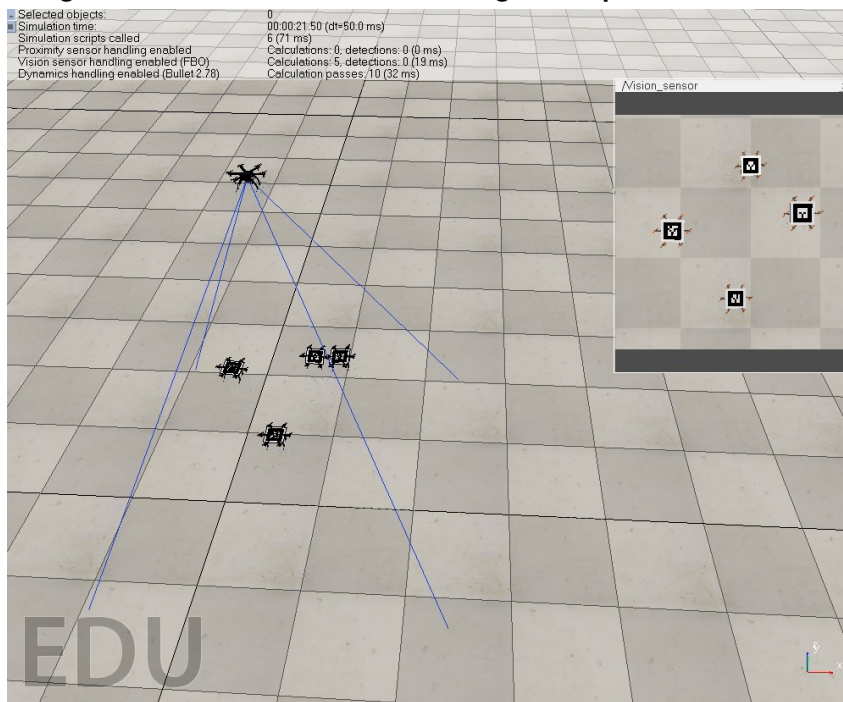
Figura 21 – Rota do VANT escravo seguindo o VANT mestre.**(a) Rota 1****(b) Rota 2****Fonte: Autoria própria.**

instruídos a abaixar 50 cm, para a altitude de 2,5 m e realizarem um quadrado de 1,4 m de lado. Os gráficos apresentados na Figura 23 representam a rota planejada, a rota realizada e a leitura da etiqueta de cada um dos VANTs escravos de forma individual.

Os erros apresentados na Tabela 2 são os erros relativos a cada um dos VANTs durante a realização do percurso. Os erros apresentados são referentes ao erro na leitura das etiquetas, tendo como base a real posição das aeronaves obtidas diretamente do CoppeliaSim. Observando os erros quadráticos médios obtidos pode-se dizer que o sistema proposto apresenta boa acurácia. E os piores casos apresentados pelo erro máximo ainda sim são toleráveis.

Como o esperado os maiores erros obtidos foram nas rotas dos VANTs que realizaram as rotas de menor altitude uma vez que esses estão mais distantes da câmera. É facilmente perceptível comparando os gráficos apresentados na Figura 23, nas figuras (c) e (d) observa-se que os valores obtidos pela leitura das etiquetas são mais ruidosos, o que prejudica o bom desempenho do algoritmo de controle tornando a movimentação das aeronaves menos precisa.

Figura 22 – VANTS escravos voando guiados pelo VANT mestre.



Fonte: Autoria própria.

5.2 AMBIENTE REAL

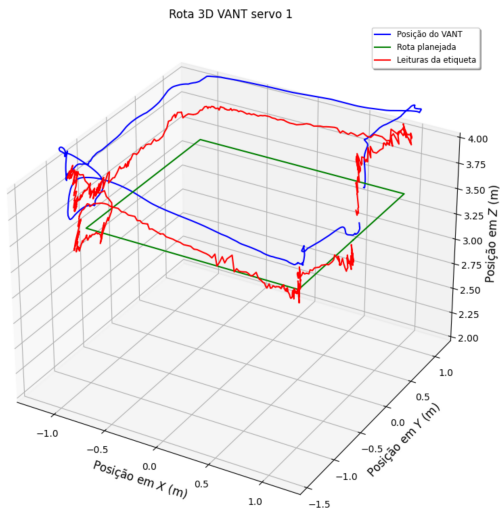
5.2.1 Aferição do erro de medida das etiquetas

O primeiro experimento em ambiente real não envolve nenhum VANT. O primeiro passo foi a aferição do erro na leitura das etiquetas, uma vez que no mundo real tem-se inúmeros fatores que interferem na leitura. Um ponto importante foi a realização das medidas com diferentes tamanho de etiquetas, já que nos próximos experimentos realizados os marcadores são menores do que os utilizados no ambiente simulado. Como foi utilizado um modelo de VANT hexacoptero na simulação foi possível utilizar etiquetas de 26 cm, o que não ocorre com as aeronaves reais. Como já ilustrados o VANT utilizado com escravo é de pequeno porte, o que permitiu a utilização de uma etiqueta de apenas 7 cm.

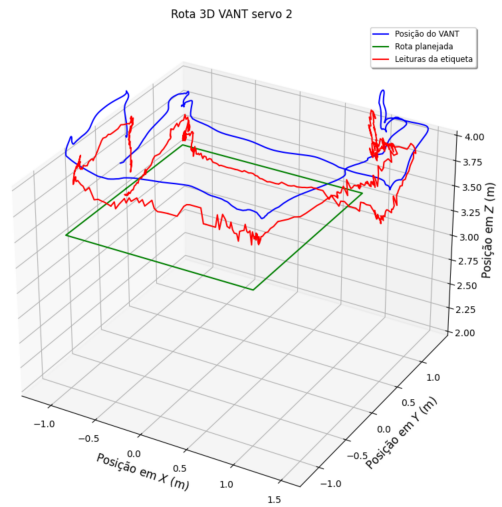
Abaixo da câmera fixa foram marcados 5 pontos no chão com as posições relativas à câmera como apresentado na Tabela 3. Três tamanhos de etiquetas diferentes foram utilizados 5 cm, 7 cm e 10 cm. O intuito desse experimento é verificar a influência do tamanho do marcador na qualidade das medidas realizadas pelo algoritmo Alvar. Foram realizadas cem medições em cada ponto que foram salvas em arquivos csv. Posteriormente foram calculados a média e desvio padrão das medidas em cada um dos pontos, informações essas apresentadas pela Tabela 4. Também foram calculados as raízes quadradas dos erros médios das medidas em cada um dos pontos, apresentados na Tabela 5

Figura 23 – Gráficos da posição dos VANTs escravos guiados pelo VANT mestre. Imagens ampliadas nos Anexos G, H, I e J

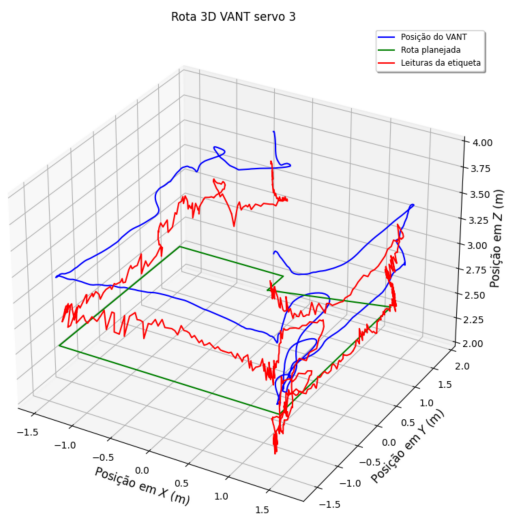
(a) Escravo 1, altitude de 3,5m



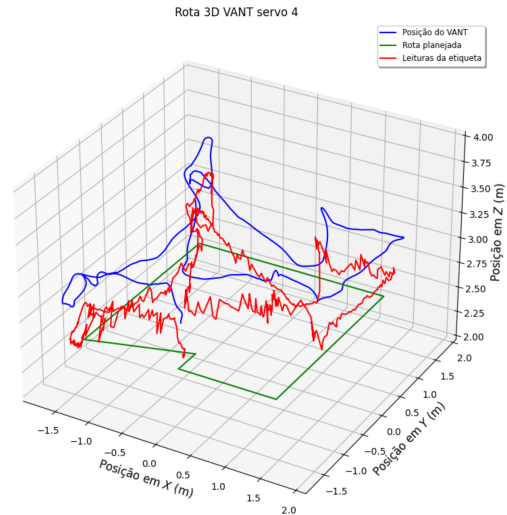
(b) Escravo 2, altitude de 3,5m



(c) Escravo 3, altitude de 2,5m



(d) Escravo 4, altitude de 2,5m



Fonte: Autoria própria.

Ao analisar os resultados apresentados em ambas as tabelas não encontra-se uma relação clara entre o tamanho da etiqueta e a qualidade das medidas coletadas. Como a diferença é muito pequena entre os valores, de desvio padrão e erro, encontrados para os diferentes tamanhos de marcadores não se pode afirmar que esse fator influencia diretamente a qualidade das informações coletadas. Esse pequeno erro pode ser oriundo por exemplo, de leves diferenças de iluminação. Apesar do experimento ter sido conduzido no mesmo ambiente sujeito às mesmas condições, a iluminação foi feita através de lâmpadas comuns as quais não possuem nenhum dispositivo para o controle preciso da luminosidade emitida.

Tabela 2 – Erros de medida das etiquetas dos múltiplos VANTs escravos.

VANT escravo 1			
	Eixo X	Eixo Y	Eixo Z
Erro médio (m)	0,004	-0,015	0,270
Desvio padrão do erro (m)	0,049	0,046	0,273
Erro máximo (m)	0,089	0,088	0,405

VANT escravo 2			
	Eixo X	Eixo Y	Eixo Z
Erro médio (m)	0,003	0,003	0,073
Desvio padrão do erro (m)	0,055	0,050	0,270
Erro máximo (m)	0,105	0,110	0,376

VANT escravo 3			
	Eixo X	Eixo Y	Eixo Z
Erro médio (m)	0,004	0,003	0,129
Desvio padrão do erro (m)	0,062	0,059	0,359
Erro máximo (m)	0,109	0,133	0,586

VANT escravo 4			
	Eixo X	Eixo Y	Eixo Z
Erro médio (m)	0,004	0,004	0,129
Desvio padrão do erro (m)	0,061	0,066	0,359
Erro máximo (m)	0,248	0,279	0,564

Tabela 3 – Posições em que os pontos foram fixados.

Ponto	X (m)	Y (m)	Z (m)
1	0,000	0,000	2,360
2	0,600	0,350	2,360
3	0,600	-0,350	2,360
4	-0,600	0,350	2,360
5	-0,600	-0,350	2,360

Diante disso, há indicação que a influência do tamanho dos marcadores está ligado ao alcance de detecção dos mesmo. Mas uma vez detectados, o tamanho não influencia na qualidade das leituras realizadas.

5.2.2 Experimento com um VANT escravo e a câmera fixa

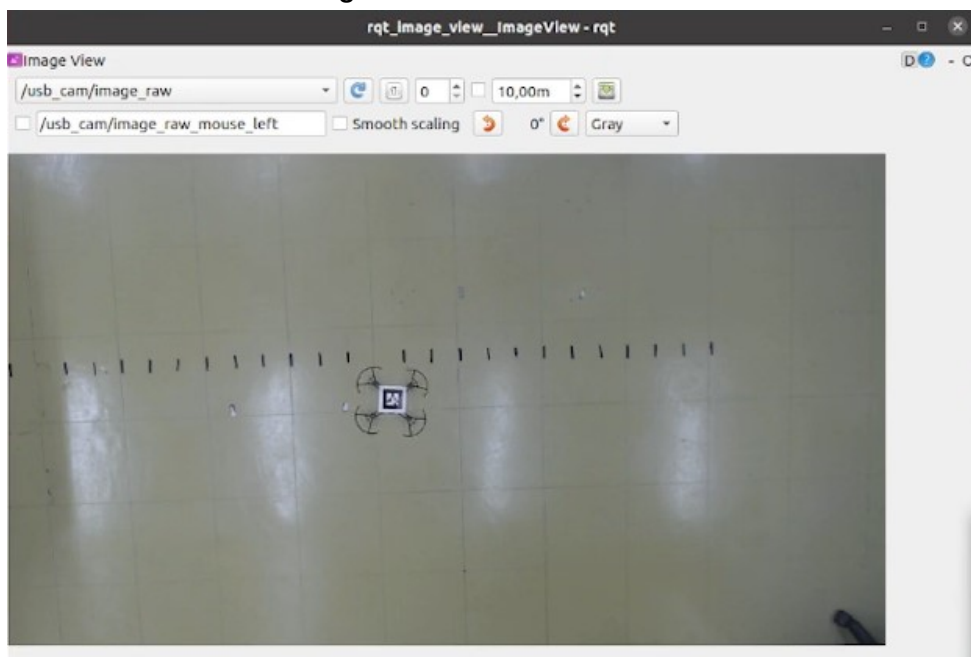
Para os experimentos conduzidos com um VANT real e a câmera fixada em um ponto fixo, uma rota em forma de um retângulo com os pontos definidos como mostrados na Tabela 6, sendo os pontos no eixo Z a distância entre o VANT e a câmera e não a altitude de voo. Na Figura 24 está ilustrada a visão capturada pela câmera, durante o voo as informações de posição foram salvas em um arquivo csv e o percurso foi reconstituído em um gráfico apresentado na Figura 25.

Tabela 4 – Média e desvio padrão das medidas.

Etiqueta de 5 cm						
Ponto	Média X (m)	Média Y (m)	Média Z (m)	Des. padrão X	Des. padrão Y	Des. padrão Z
1	0,004	-0,006	2,894	0,000	0,000	0,003
2	0,645	0,328	2,881	0,001	0,001	0,006
3	0,609	-0,391	2,859	0,001	0,001	0,006
4	-0,600	0,386	2,860	0,001	0,001	0,006
5	-0,641	-0,323	2,855	0,003	0,001	0,012

Etiqueta de 7 cm						
Ponto	Média X (m)	Média Y (m)	Média Z (m)	Des. padrão X	Des. padrão Y	Des. padrão Z
1	-0,003	0,000	2,864	0,000	0,000	0,002
2	0,630	0,330	2,859	0,001	0,001	0,006
3	0,603	-0,390	2,829	0,001	0,001	0,006
4	-0,607	0,386	2,843	0,001	0,000	0,003
5	-0,652	-0,325	2,860	0,001	0,000	0,003

Etiqueta de 10 cm						
Ponto	Média X (m)	Média Y (m)	Média Z (m)	Des. padrão X	Des. padrão Y	Des. padrão Z
1	0,004	0,002	2,867	0,000	0,000	0,002
2	0,640	0,328	2,881	0,001	0,000	0,003
3	0,617	-0,394	2,889	0,001	0,001	0,005
4	-0,612	0,377	2,862	0,000	0,000	0,002
5	-0,646	-0,336	2,877	0,000	0,000	0,002

Figura 24 – Visão da câmera.

Fonte: Autoria própria.

Tabela 5 – Raiz quadrado do erro médio das medidas.

Etiqueta de 5 cm			
Ponto	Raiz quadrada do erro médio em X (m)	Raiz quadrada do erro médio em Y (m)	Raiz quadrada do erro médio em Z (m)
1	0,004	0,006	0,534
2	0,045	0,022	0,521
3	0,009	0,041	0,499
4	0,001	0,036	0,500
5	0,041	0,027	0,495

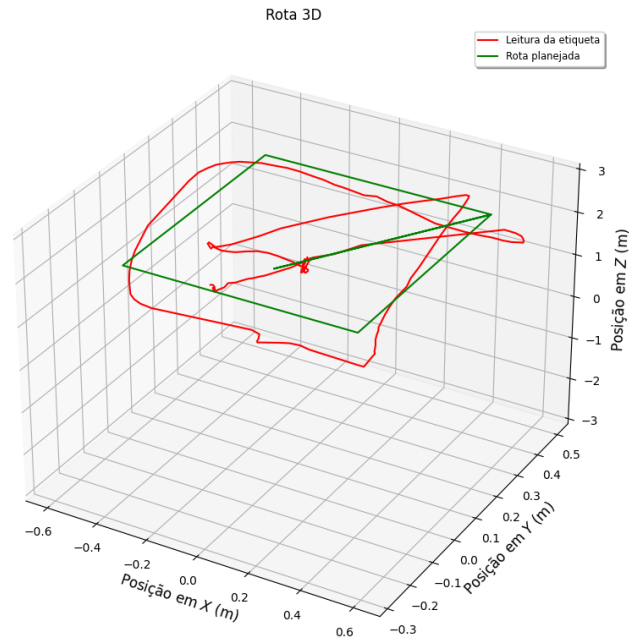
Etiqueta de 7 cm			
Ponto	Raiz quadrada do erro médio em X (m)	Raiz quadrada do erro médio em Y (m)	Raiz quadrada do erro médio em Z (m)
1	0,003	0,000	0,504
2	0,030	0,020	0,499
3	0,003	0,040	0,469
4	0,007	0,036	0,483
5	0,052	0,025	0,500

Etiqueta de 10 cm			
Ponto	Raiz quadrada do erro médio em X (m)	Raiz quadrada do erro médio em Y (m)	Raiz quadrada do erro médio em Z (m)
1	0,004	0,002	0,507
2	0,040	0,022	0,521
3	0,017	0,044	0,529
4	0,012	0,027	0,502
5	0,046	0,014	0,517

Tabela 6 – Pontos da rota planejada.

Ponto	X (m)	Y (m)	Z (m)
1	0.00	0.00	2.00
2	0.45	0.45	2.00
3	-0.45	0.45	2.00
4	-0.45	-0.15	2.00
5	0.45	-0.15	2.00
6	0.45	0.45	2.00
7	0.00	0.00	2.00

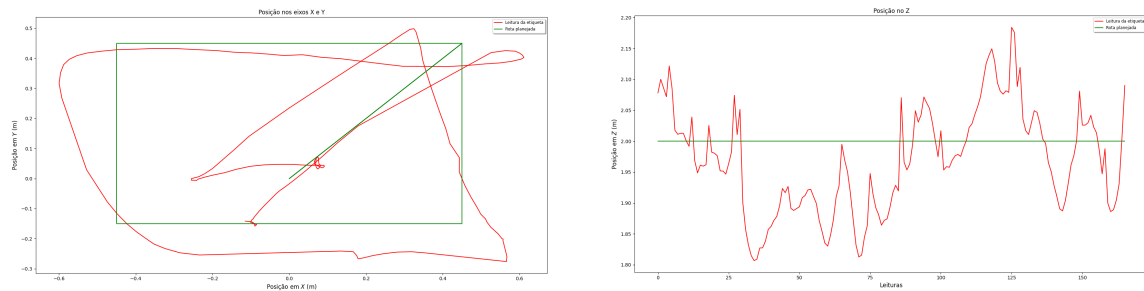
Figura 25 – Rota realizada pelo VANT. Imagem ampliada no Anexo K



Fonte: Autoria própria.

Para melhorar o entendimento e auxiliar a visualização da rota executada pelo VANT, foram gerados os gráficos da posição nos eixos X e Y e outro gráfico da posição no eixo Z apresentados na Figura 24

Figura 26 – Gráficos da posição do VANT. Imagens ampliadas nos Anexos L e M
(a) Gráfico da posição nos eixos X e Y. **(b) Gráfico da posição no eixo Z.**



Fonte: Autoria própria.

Neste experimento foi possível obter indicações de que o sistema de controle desenvolvido neste trabalho funciona de forma satisfatória. O próximo passo foi a realização de os testes com a arquitetura utilizando um VANT mestre e um VANT escravo.

5.2.3 Experimento com um VANT mestre e um VANT escravo

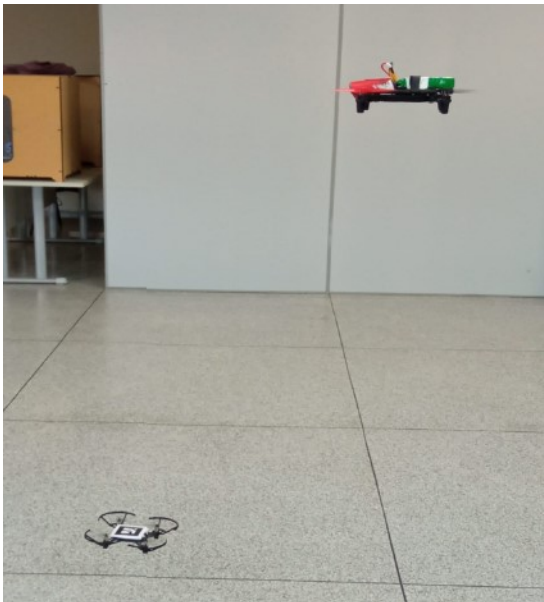
Esse último experimento foi realizado para verificar a viabilidade da arquitetura mestre-escravo. Como VANT mestre foi utilizado o Bebop, ele foi controlado manualmente a partir de

uma estação base. O Tello com a etiqueta em seu topo foi utilizado como VANT escravo. O objetivo era fazer com que a aeronave escrava mantivesse sua posição relativa à outra aeronave, no ponto (0,00; 0,00; -1,00), e seguido o VANT mestre enquanto ele se movimentava. Pode-se observar imagens dos testes realizados na Figura 27.

Os testes foram conduzidos de forma que o VANT mestre, controlado manualmente, foi posicionado acima do VANT escravo e então o algoritmo de controle da aeronave escrava começou a ser executado fazendo com que ela siga a aeronave mestre. O VANT mestre foi movimentado com velocidade de 0,1 m/s aproximadamente e o voo teve duração próxima à um minuto.

Figura 27 – Imagens do experimento da arquitetura com VANT mestre e escravo.

(a) VANT escravo mantendo a posição.



(b) Visão da câmera do VANT mestre.



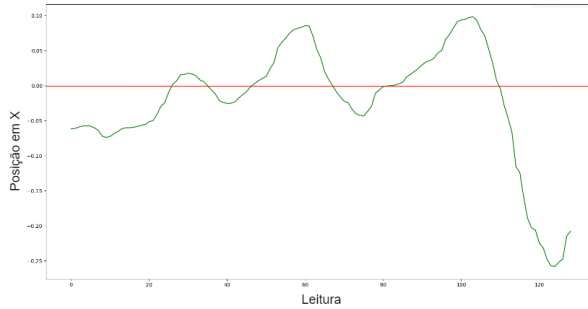
Fonte: Autoria própria.

Durante os testes os dados de posição do VANT escravo foram coletados e salvos em um arquivo csv. Esses dados serviram para a confecção dos gráficos apresentados pela Figura 28, eles ilustram a posição nos eixos X, Y e Z da aeronave escrava ao longo do tempo. Nos gráficos as linhas vermelhas representam o ponto objetivo ao qual o VANT escravo tenta alcançar e as linhas verdes representam as medições da posição relativa entre os VANTs providas das leituras das etiquetas. Em um cenário ideal as linhas seriam sobrepostas, mas é possível observar que o sistema de controle da aeronave escrava obteve um desempenho adequado estabilizando o Tello próximo ao ponto objetivo, tendo um erro máximo de pouco mais de 20 cm aproximadamente.

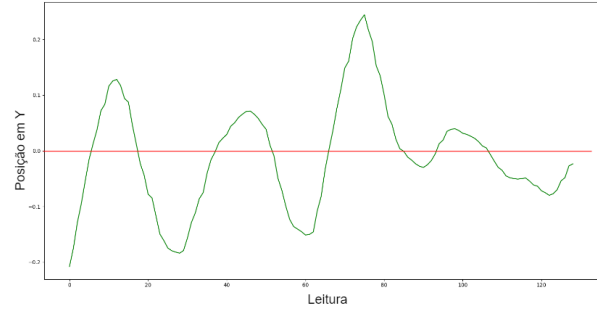
Apesar das dificuldades encontradas durante os testes (por exemplo a perda de quadros na transmissão de imagens entre o Bebop e a estação base) pode-se ter bons indicativos de que essa arquitetura mestre-escravo é viável, algumas variações aconteceram mas o sistema

Figura 28 – Gráficos da posição do VANT escravo em relação ao mestre. Imagens ampliadas nos Anexos N, O e P

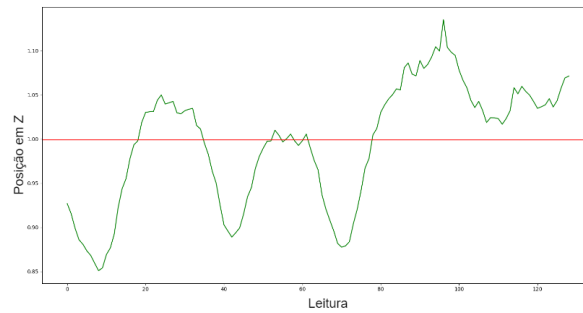
(a) Posição no eixo X.



(b) Posição no eixo Y.



(c) Posição no eixo Z.



Fonte: Autoria própria.

de controle conseguiu manter o VANT escravo estável próximo ao ponto objetivo definido. Os resultados obtidos foram satisfatórios e são um bom indicativo.

6 CONCLUSÕES

Existem diferentes maneiras de criar um sistema de inspeção automático. Neste projeto foi proposto a utilização de múltiplos VANTS trabalhando de forma cooperativa para coletar imagens. Especificamente, uma arquitetura contendo um VANT mestre que usa a identificação de tags em VANTS escravos para estimar o posicionamento destes. Para isto, foi elaborado um ambiente simulado para a realização de testes do modelo. A partir disso, foram feitos testes reais para validar o funcionamento da arquitetura proposta.

O uso de tags para estimar a posição ao invés do uso de GPS trazem alguns benefícios. Para se obter uma qualidade razoável na coleta de imagens, os VANTS devem estar relativamente próximos da estrutura a ser inspecionada. Assim, a qualidade do sinal de GPS pode ser comprometida, já que essas estruturas geralmente são grandes ou compostas de metal. Nesse cenário, o uso de tags para estimar o posicionamento dos VANTS escravos, responsáveis pela coleta de imagens, não sofre interferência dessas estruturas.

Os experimentos indicam que a arquitetura proposta é uma forma confiável de criar um sistema de controle baseado em visão para VANTS colaborativos. Os resultados apresentam pequenas medições de erros em simulação. Em experimentos do mundo real, o ambiente e condições de hardware impactam na estabilidade do controle, principalmente nos voos mestre-servo, devido às limitações de captura de imagem do drone Bebop.

A qualidade das imagens capturadas pela câmera é uma limitação para este trabalho. Nos segundo experimento realizado em ambiente real, a *Webcam* oferece restrições como tempo de correção de brilho lento, baixa transferência de dados para o PC devido à conexão *USB 2.0* e nenhum mecanismo de foco automático. Essas limitações impactam na captura de quadros e no processamento das etiquetas AR-Tag, sendo possível avaliar o uso de câmeras profissionais para aumentar a qualidade da posição dos dados e a robustez do sistema de posicionamento em ambientes internos. Além da câmera do drone Bebop adquirir imagens *Full HD*, o pacote *bebop_autonomy* transmite apenas quadros de resolução 640x480 para a estação base, o que aumenta o erro quando se calcula a posição do marcador. Além disso, os voos de seguimento executados apresentaram uma operação adequada quando a captura de marcadores fornece bons dados de posição ao algoritmo de controle PID.

O deslocamento da câmera causado pela inclinação VANT mestre durante as mudanças de rota gera o erro de cálculo da posição da etiqueta. Ainda assim, o sistema de estabilização de imagem do Bebop foi capaz de compensar isso, evitando a perda de controle da aeronave. Além disso, o tamanho pequeno da etiqueta exige manter o VANT servo em voo próximo com o mestre. Por causa disso, a área de captura da câmera é reduzida, de modo que o drone mestre às vezes perde a visibilidade do servo e o controle de posição é perdido. Essa dificuldade pode ser minimizada se um drone servo for capaz de carregar uma etiqueta maior do tamanho que foi utilizado.

Alguns quadros de imagem enviados pelo drone Bebop para a estação base foram perdidos. Estes causam problemas na geração de dados de posição pelo pacote *ar_track_alvar* que afetam diretamente o bom funcionamento do algoritmo de controle de voo do VANT, mas mesmo com essa adversidade o sistema se mostrou robusto o suficiente para funcionar, como apresentado.

Problemas práticos devem ser resolvidos para permitir a aplicação deste método em operações reais de robôs, principalmente em ambientes externos não estruturados. Ainda assim, é possível considerar que uma primeira prova de conceito da arquitetura colaborativa obteve resultados satisfatórios.

6.1 TRABALHOS FUTUROS

O uso de uma câmera de melhor resolução e qualidade de imagem, arranjos de captura de etiquetas maiores e a fusão de multisensores é uma maneira interessante de obter melhores resultados para a arquitetura proposta.

Trabalhos futuros estão planejados para investigar novos sistemas de obtenção de dados de posicionamento visual e arranjos que possam fornecer melhores dados para as aplicações, como etiquetas de luz ativas, por exemplo. Além disso, serão investigados problemas práticos do mundo real que podem ser resolvidos usando a arquitetura cooperativa proposta.

REFERÊNCIAS

- AL-KAFF, A. *et al.* Survey of computer vision algorithms and applications for unmanned aerial vehicles. **Expert Systems with Applications**, Elsevier, v. 92, p. 447–463, 2018.
- AMIR, M. Y.; ABBASS, V. Modeling of quadrotor helicopter dynamics. *In: IEEE. 2008 International Conference on Smart Manufacturing Application*. [S.l.], 2008. p. 100–105.
- BERGER, G. S. Método para projetar arquiteturas de sensores para detecção de obstáculos em veículos aéreos não tripulados dedicados à inspeção de torres de alta tensão. 2020.
- BOUDJIT, K.; LARBES, C. Detection and implementation autonomous target tracking with a quadrotor ar. drone. *In: IEEE. 2015 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*. [S.l.], 2015. v. 2, p. 223–230.
- CANTIERI, A. *et al.* Cooperative uav–ugv autonomous power pylon inspection: An investigation of cooperative outdoor vehicle positioning architecture. **Sensors**, v. 20, n. 21, 2020. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/20/21/6384>.
- FINLAND, V. T. R. C. of. **Augmented Reality / 3D Tracking**. 2017. Disponível em: <http://virtual.vtt.fi/virtual/proj2/multimedia/>.
- GARCIA, E. *et al.* The evolution of robotics research. **IEEE Robotics & Automation Magazine**, v. 14, n. 1, p. 90–103, 2007.
- GUIZZO, E. **What Is a Robot? Top roboticists explain their definition of robot**. 2018. Disponível em: <https://robots.ieee.org/learn/what-is-a-robot/>.
- HAMILTON, W. R. Xi. on quaternions; or on a new system of imaginaries in algebra. **The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science**, Taylor & Francis, v. 33, n. 219, p. 58–60, 1848.
- HASSANALIAN, M.; ABDELKEFI, A. **Classifications, applications, and design challenges of drones: A review**. [S.l.]: Elsevier Ltd, 2017. 99-131 p.
- HINAS, A.; ROBERTS, J. M.; GONZALEZ, F. Vision-based target finding and inspection of a ground target using a multirotor uav system. **Sensors**, MDPI, v. 17, n. 12, p. 2929, 2017.
- HOOD, S. *et al.* Bird's eye view: Cooperative exploration by ugv and uav. *In: IEEE. 2017 International Conference on Unmanned Aircraft Systems (ICUAS)*. [S.l.], 2017. p. 247–255.
- IDRIES, A. *et al.* Challenges of developing uav applications: A project management view. **IEOM 2015 - 5th International Conference on Industrial Engineering and Operations Management, Proceeding**, 04 2015.
- KARDASZ, P.; DOSKOCZ, J. Drones and possibilities of their using. **Journal of Civil & Environmental Engineering**, OMICS Publishing Group, v. 6, 2016.
- KAYHANI, N. *et al.* Improved tag-based indoor localization of uavs using extended kalman filter. *In: Proceedings of the ISARC. International Symposium on Automation and Robotics in Construction, Banff, AB, Canada*. [S.l.: s.n.], 2019. p. 21–24.
- KNOSPE, C. Pid control. **IEEE Control Systems Magazine**, IEEE, v. 26, n. 1, p. 30–31, 2006.
- KRUPKE, D. *et al.* Flexible modular robotic simulation environment for research and education. *In: ECMS*. [S.l.: s.n.], 2012. p. 243–249.

LEE, H.-S. *et al.* Development of multiple uav collaborative driving systems for improving field phenotyping. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 22, p. 1423, 2 2022. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/22/4/1423>.

LU, J. *et al.* **Multi-machines and Multi-tasks Scheduling for UAV Power Inspection in Smart Grid**. 2022. 1579-1587 p.

MARENGONI, M.; STRINGHINI, D. **Introdução à Visão Computacional usando OpenCV**. 2009.

MARINHO, C. A. *et al.* In-service flares inspection by unmanned aerial vehicles (uavs). *In: . [S.l.: s.n.]*, 2012.

MONAJJEMI, M. **ROS Driver for Parrot Bebop Drone**. 2015. Disponível em: <https://bebop-autonomy.readthedocs.io/en/latest/>.

MUSKARDIN, T. *et al.* A novel landing system to increase payload capacity and operational availability of high altitude long endurance uav. *In: 2016 International Conference on Unmanned Aircraft Systems (ICUAS)*. [S.l.: s.n.], 2016. p. 495–504.

OLSON, E. **AprilTags Visual Fiducial System**. 2018. Disponível em: <https://april.eecs.umich.edu/software/apriltag>.

POPPER, B. **PARROT BEBOP DRONE REVIEW: AMBITIOUS, UNIQUE, BUT DANGEROUSLY INCONSISTENT**. 2015. Disponível em: <https://www.theverge.com/2015/6/5/8727399/parrot-bebop-drone-review>.

PRISACARIU, V. The history and the evolution of uavs from the beginning till the 70 s. *In: . [S.l.: s.n.]*, 2017.

PYO, Y. *et al.* **ROS Robot Programming**. 1st. ed. [S.l.]: ROBOTIS Co.,Ltd., 2017. ISBN 979-11-962307-1-5.

ROBOTICS, L. C. **CoppeliaSim User Manual**. 2019. Disponível em: <https://www.coppeliarobotics.com/helpFiles/>.

ROHMER, E.; SINGH, S. P.; FREESE, M. V-rep: A versatile and scalable robot simulation framework. *In: IEEE. 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. [S.l.], 2013. p. 1321–1326.

RUI, M. *et al.* Multi-uav power inspection task planning technology based on deep reinforcement learning. **Computer and Modernization**, Computer and Modernization, v. 0, n. 01, p. 98–102, 2022.

SICILIANO, B.; KHATIB, O.; KRÖGER, T. **Springer handbook of robotics**. [S.l.]: Springer, 2008. v. 200.

SILTANEN, S. Theory and applications of marker-based augmented reality: Licentiate thesis. 2012.

SKULSTAD, R. *et al.* Autonomous net recovery of fixed- wing uav with single-frequency carrier-phase differential gnss. **IEEE Aerospace and Electronic Systems Magazine**, v. 30, p. 18–27, 2015.

WANG, S.; HU, T. Ros-gazebo supported platform for tag-in-loop indoor localization of quadcopter. *In: CHEN, W. et al. (Ed.). Intelligent Autonomous Systems 14*. Cham: Springer International Publishing, 2017. p. 185–197. ISBN 978-3-319-48036-7.

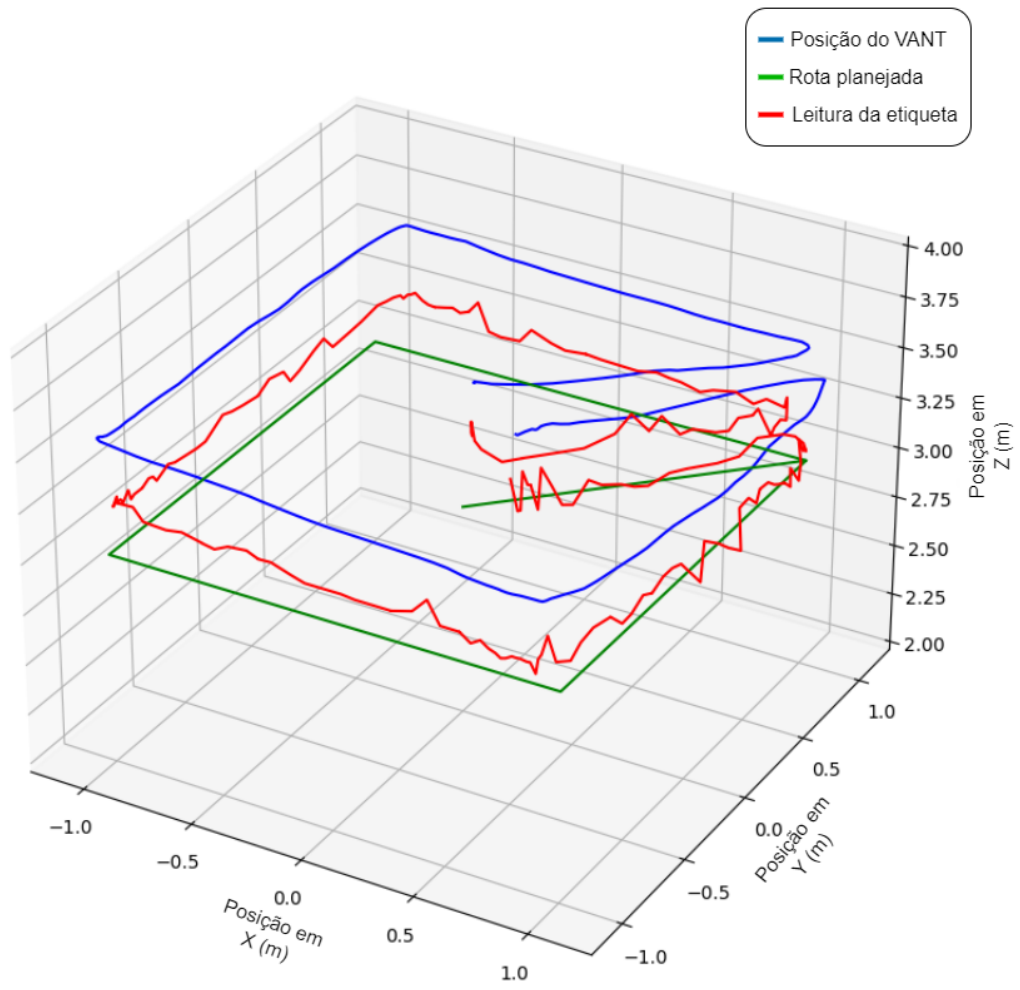
WHITE, J. K. P.; INGALLS, R. G. The basics of simulation. *In: . [S.l.: s.n.], 2018.*

XU, A. **Tello driver**. 2018. Disponível em: https://github.com/anqixu/tello_driver.

ZIEGLER, J.; NICHOLS, N. B. **Optimum Settings for Automatic Controllers**. 1942.

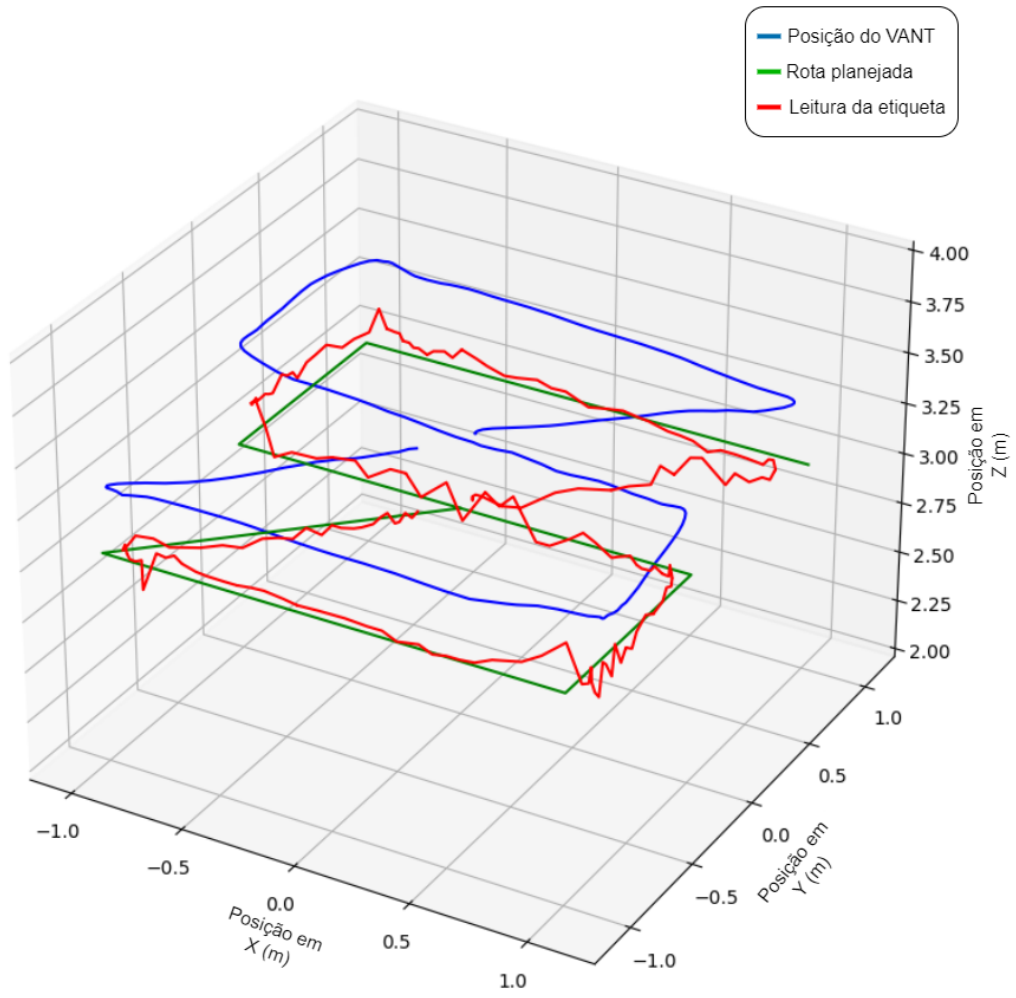
ZIMMERMANN, F. *et al.* Precise positioning of uavs-dealing with challenging rtk-gps measurement conditions during automated uav flights. **ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences**, v. 4, 2017.

ANEXO A – Gráfico da posição do VANT Figura 19 (a)



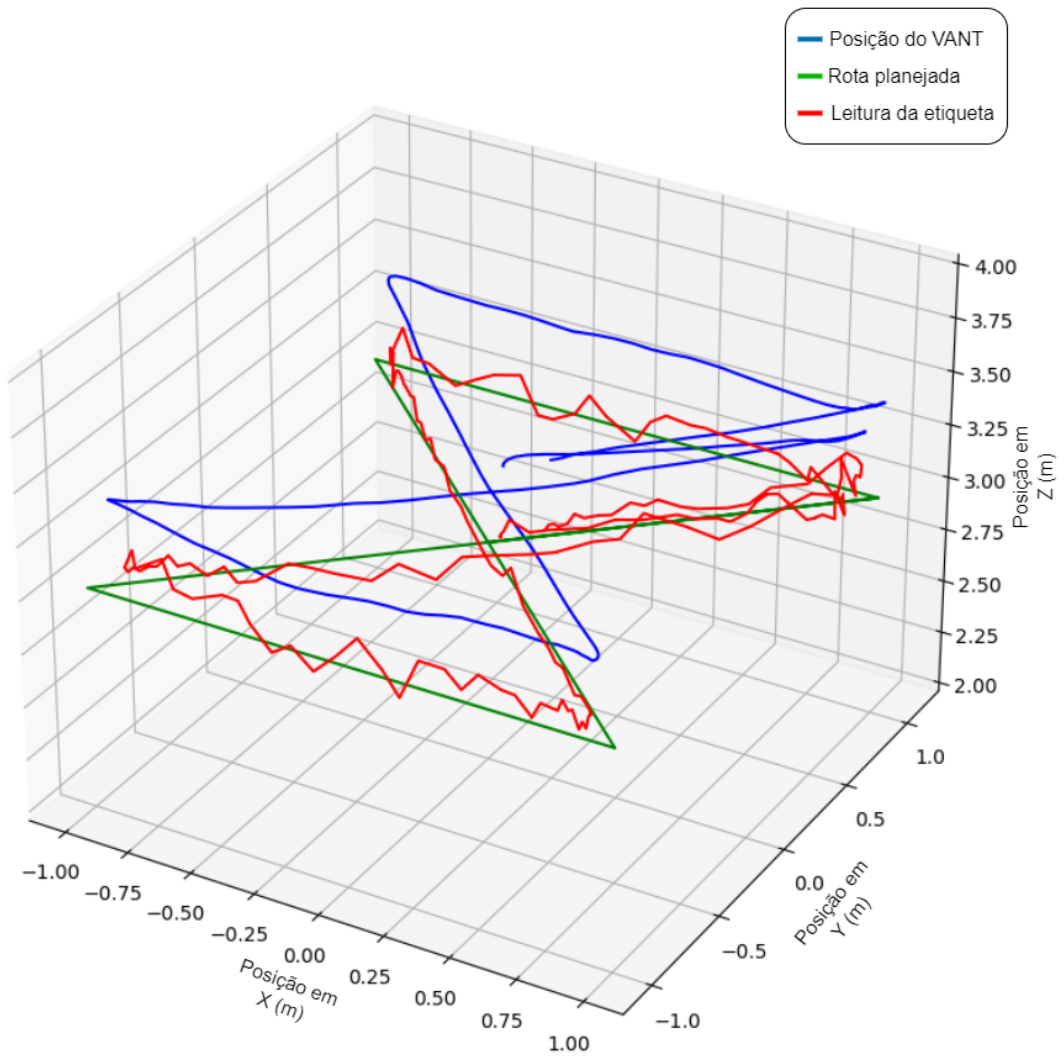
Fonte: Autoria própria.

ANEXO B – Gráfico da posição do VANT Figura 19 (b)



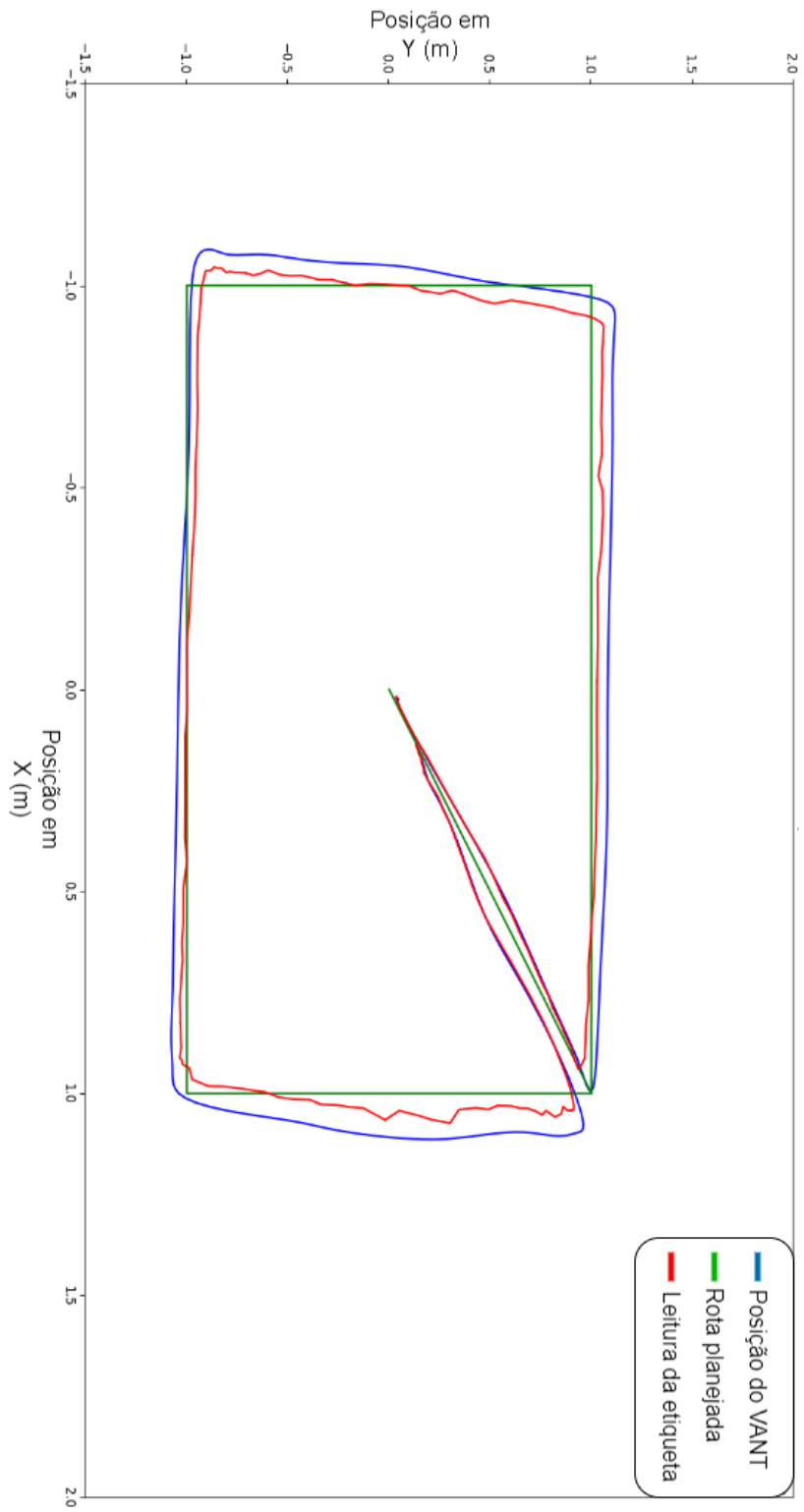
Fonte: Autoria própria.

ANEXO C – Gráfico da posição do VANT Figura 19 (c)



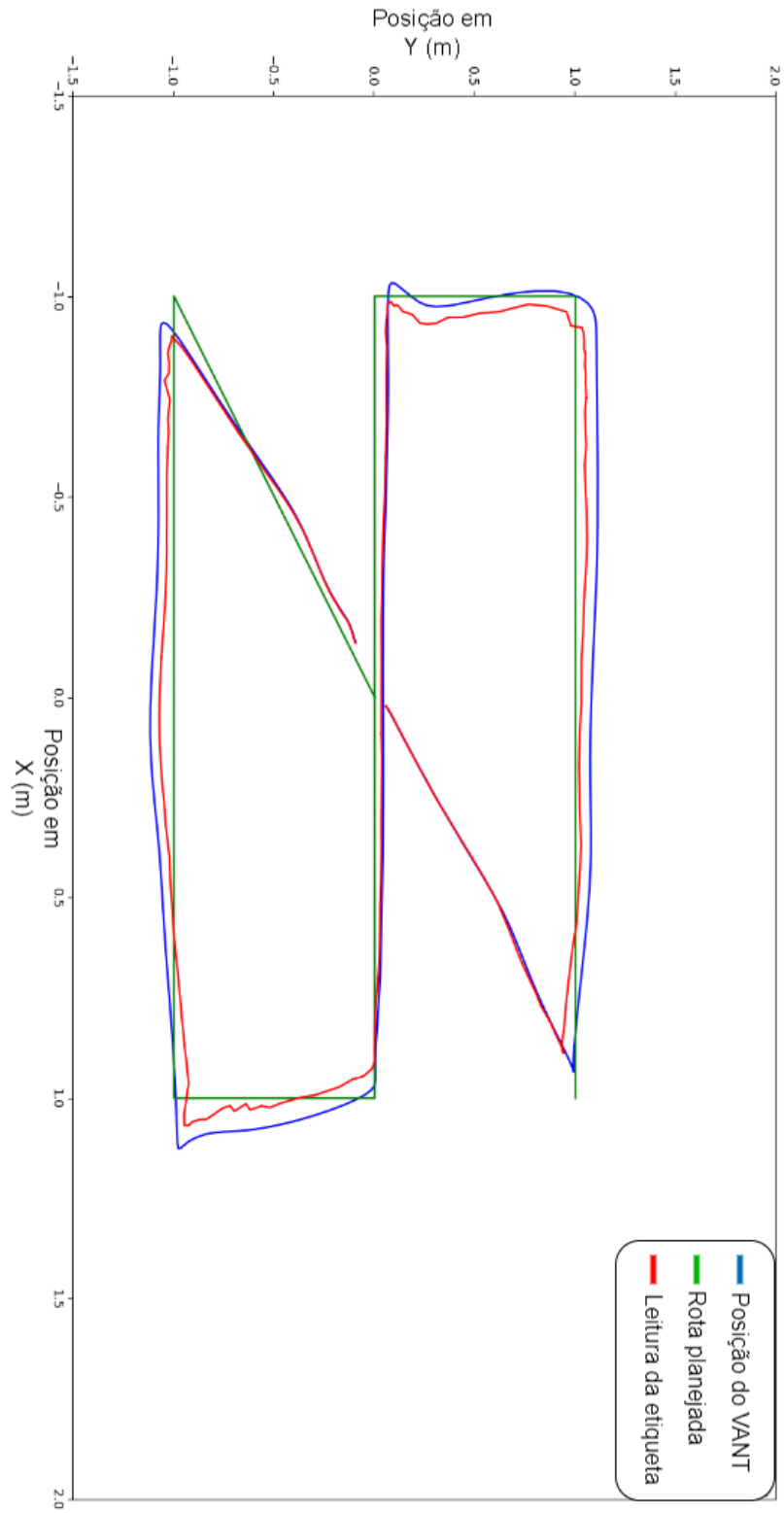
Fonte: Autoria própria.

ANEXO D – Gráfico da posição do VANT Figura 20 (a)



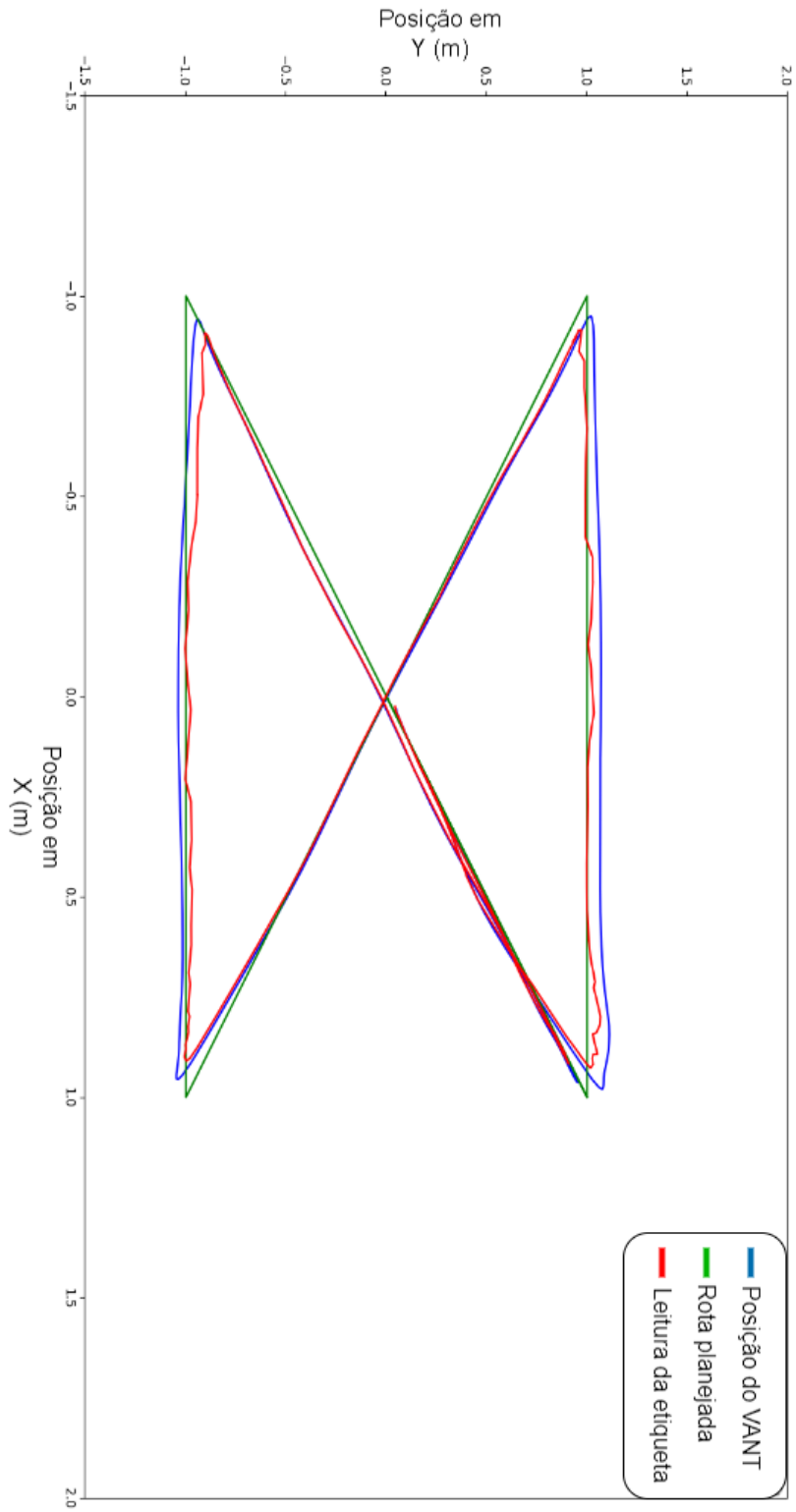
Fonte: Autoria própria.

ANEXO E – Gráfico da posição do VANT Figura 20 (b)



Fonte: Autoria própria.

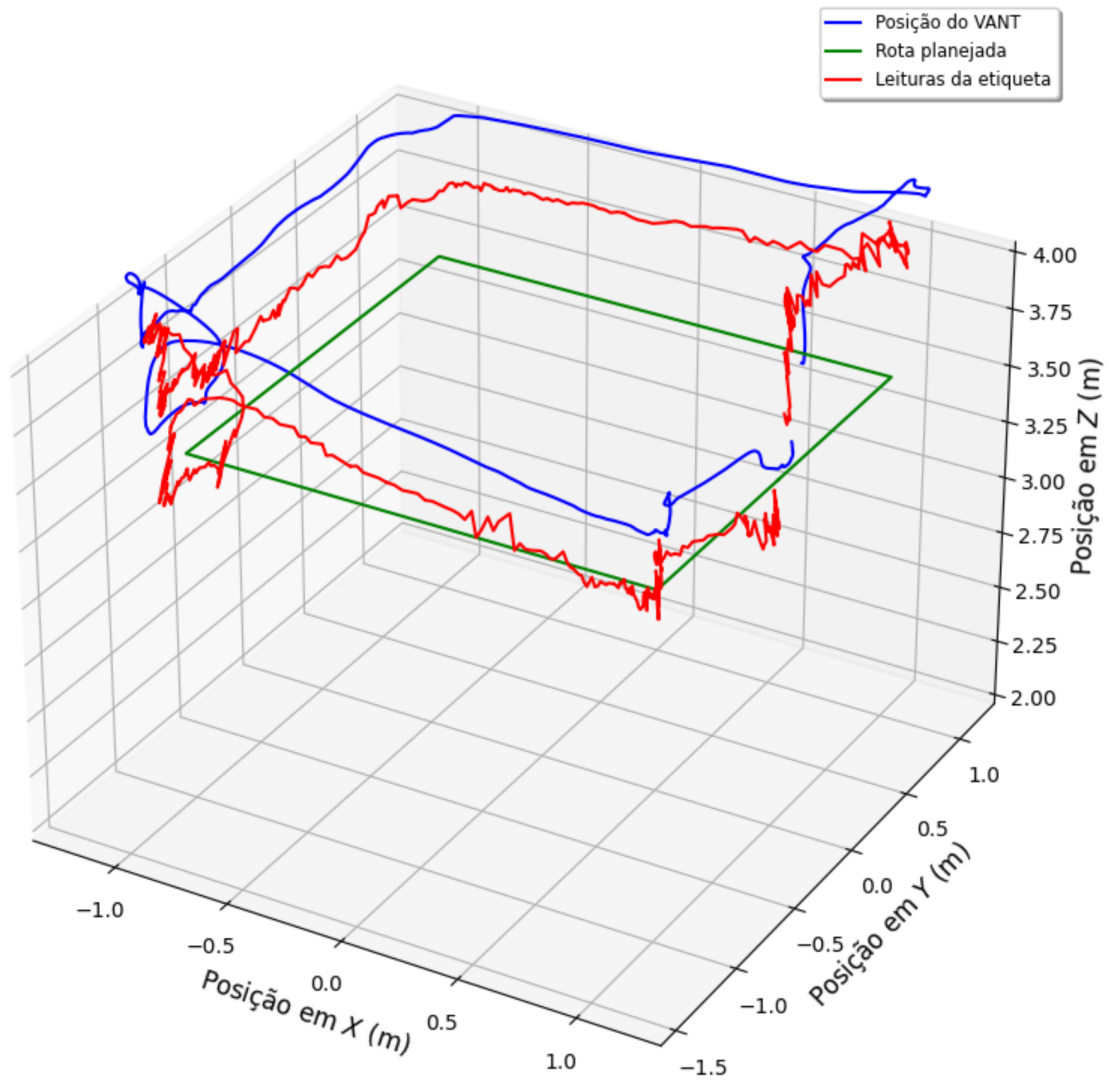
ANEXO F – Gráfico da posição do VANT Figura 20 (c)



Fonte: Autoria própria.

**ANEXO G – Gráficos da posição dos VANTs escravos guiados pelo VANT
mestre Figura 23 (a)**

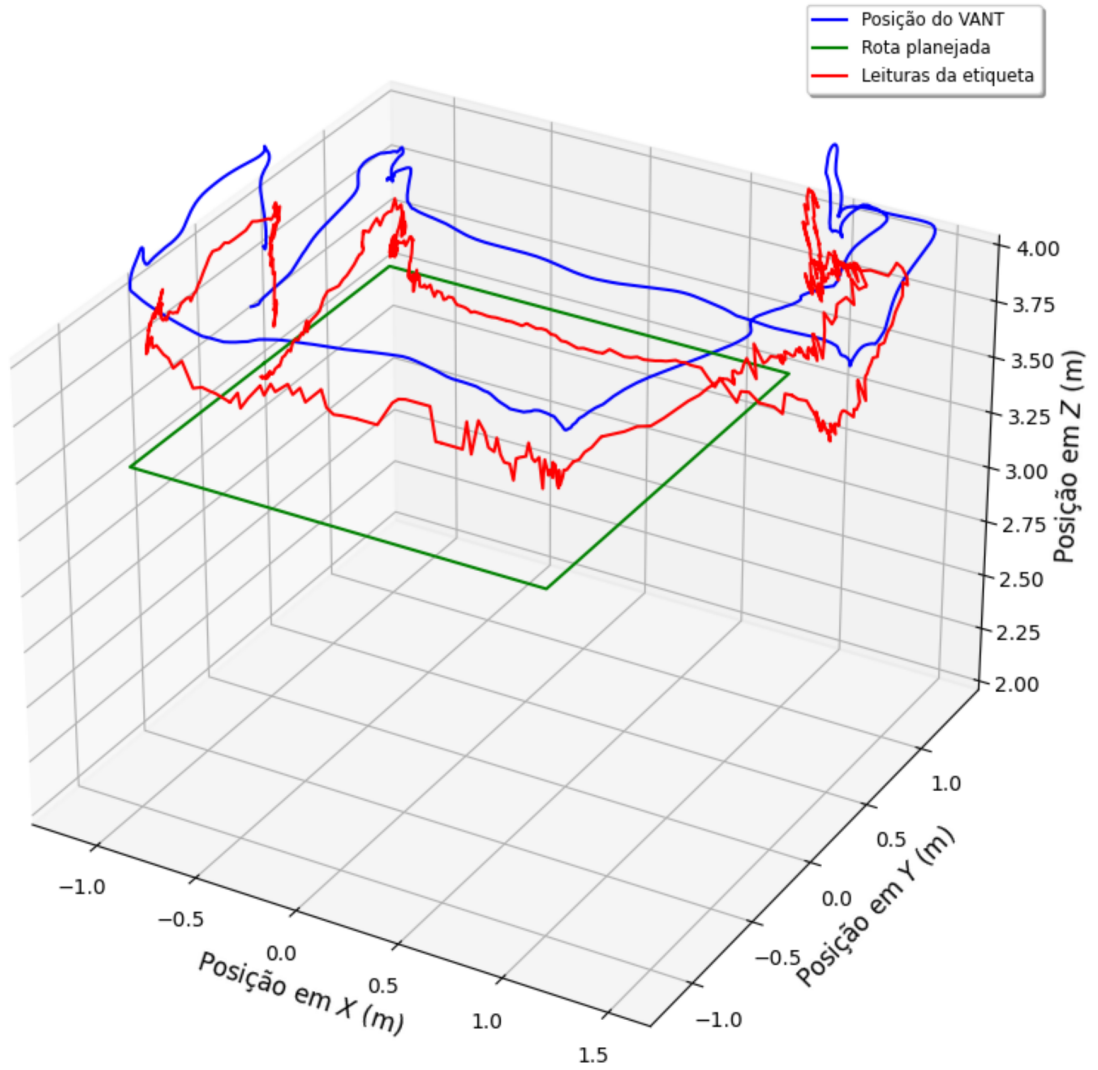
Rota 3D VANT servo 1



Fonte: Autoria própria.

**ANEXO H – Gráficos da posição dos VANTs escravos guiados pelo VANT
mestre Figura 23 (b)**

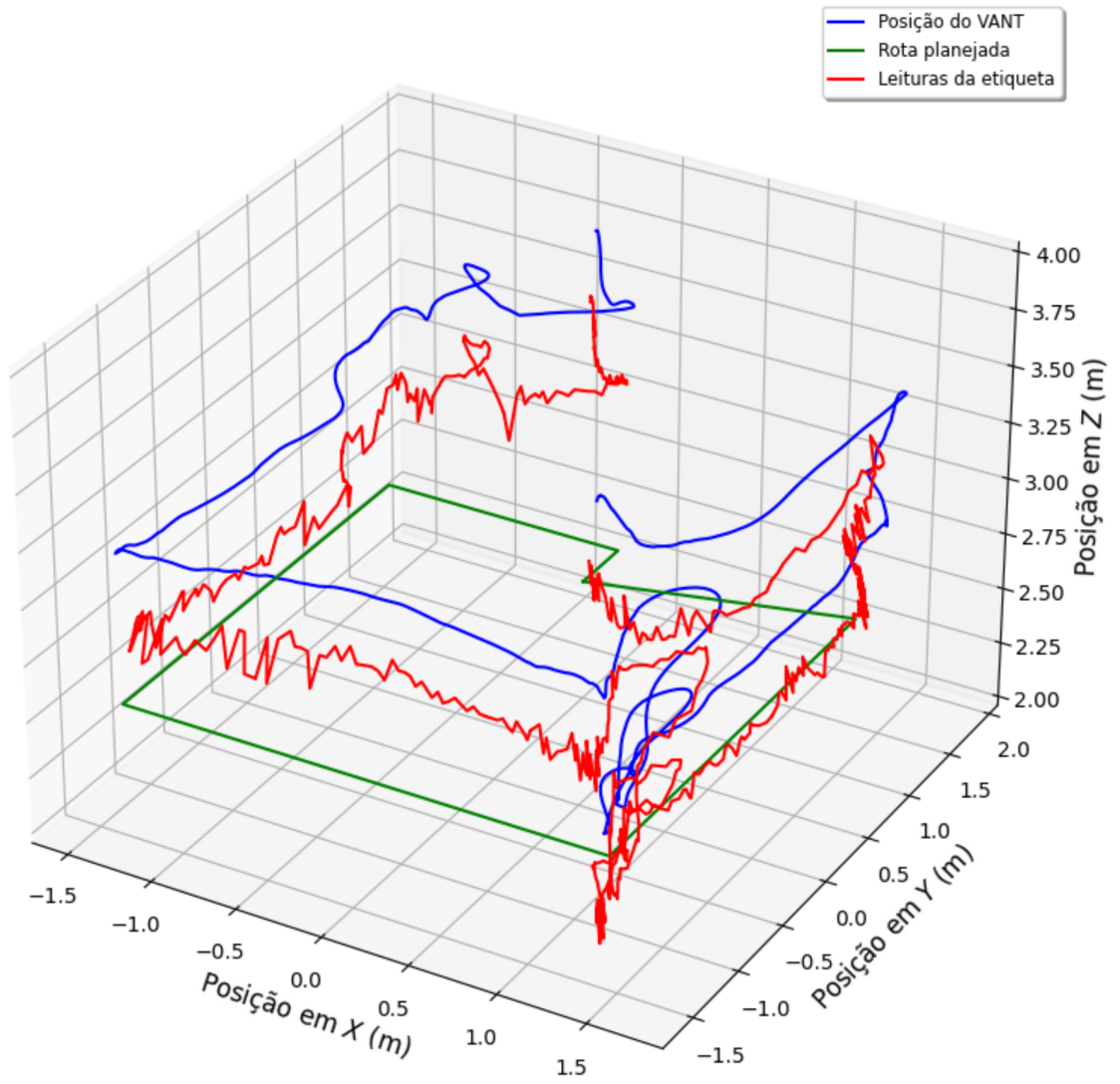
Rota 3D VANT servo 2



Fonte: Autoria própria.

**ANEXO I – Gráficos da posição dos VANTs escravos guiados pelo VANT
mestre Figura 23 (c)**

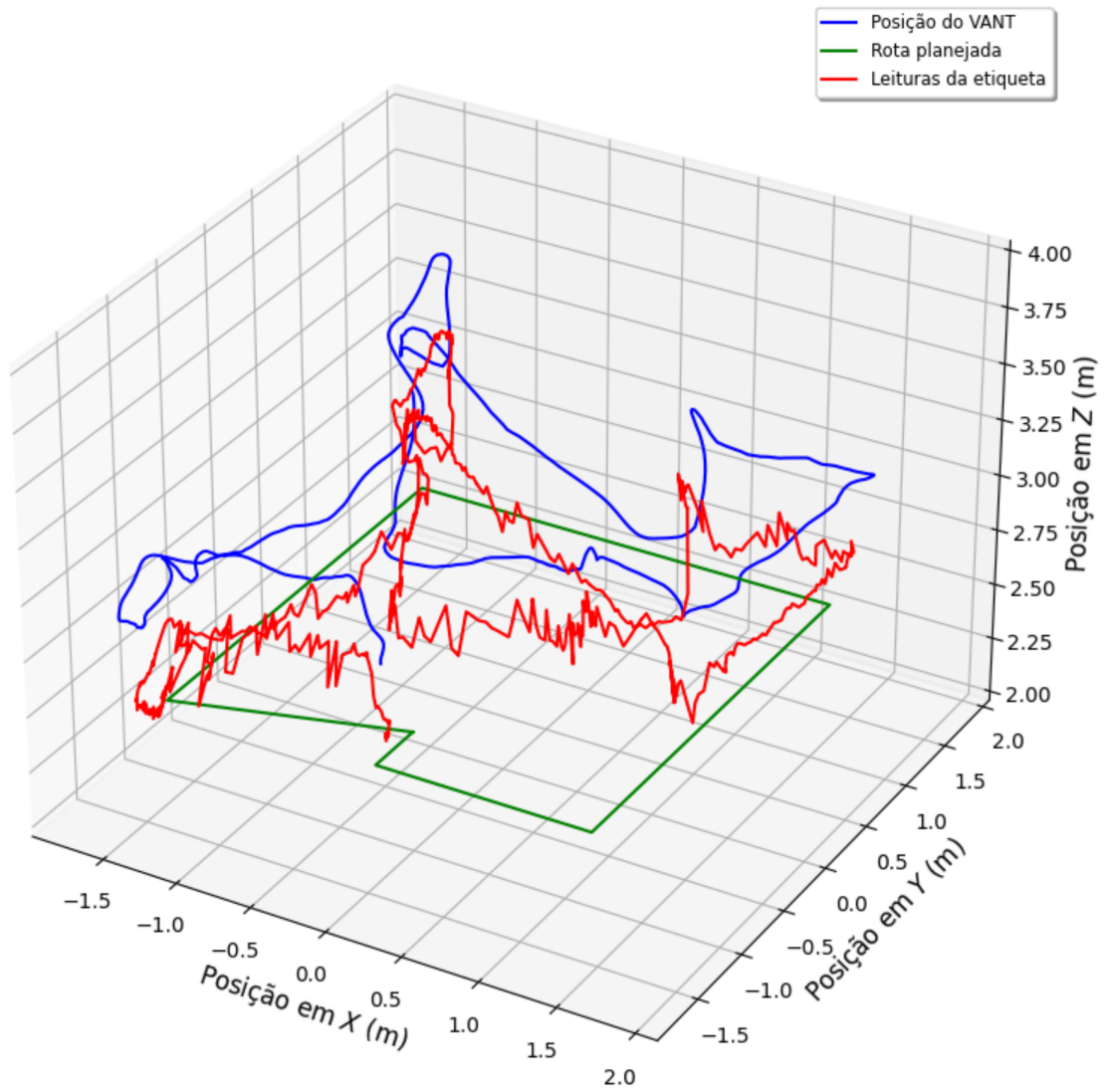
Rota 3D VANT servo 3



Fonte: Autoria própria.

**ANEXO J – Gráficos da posição dos VANTs escravos guiados pelo VANT
mestre Figura 23 (d)**

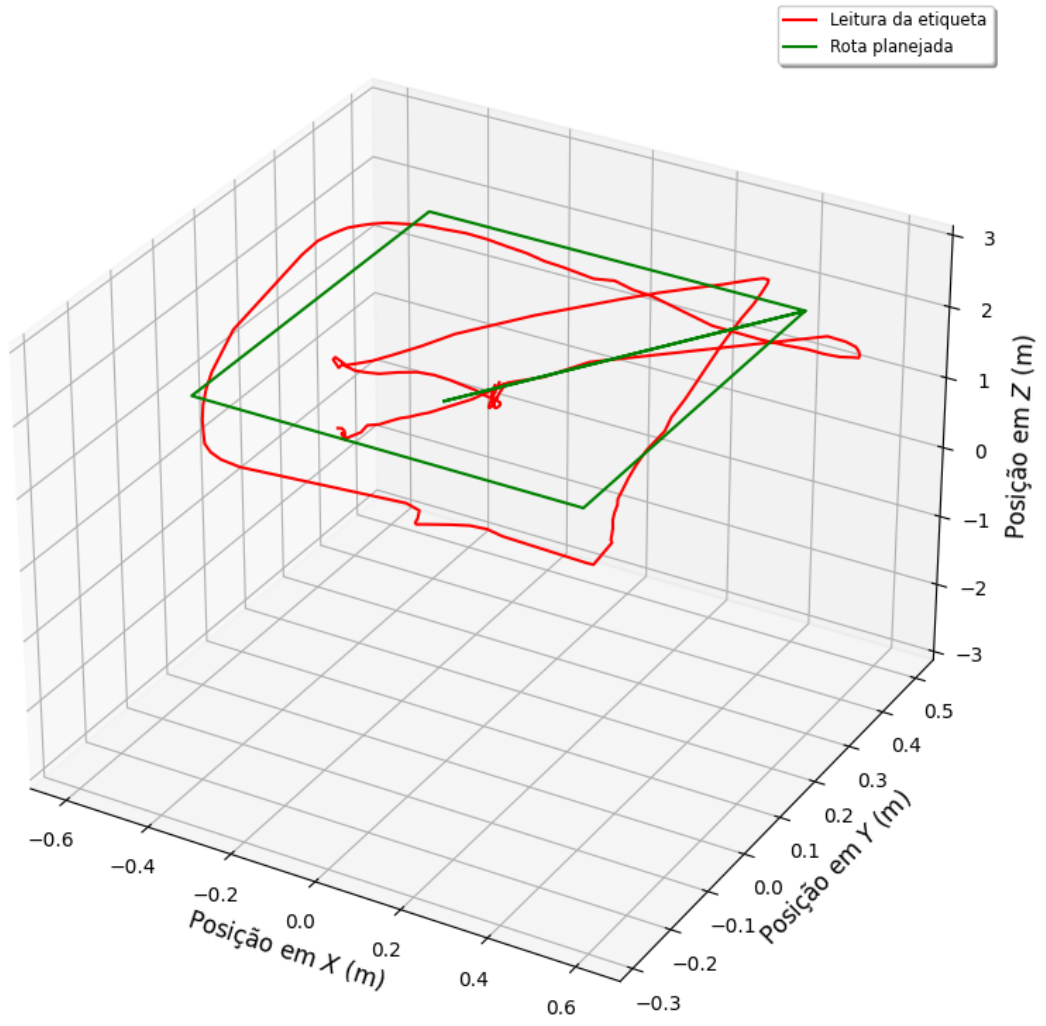
Rota 3D VANT servo 4



Fonte: Autoria própria.

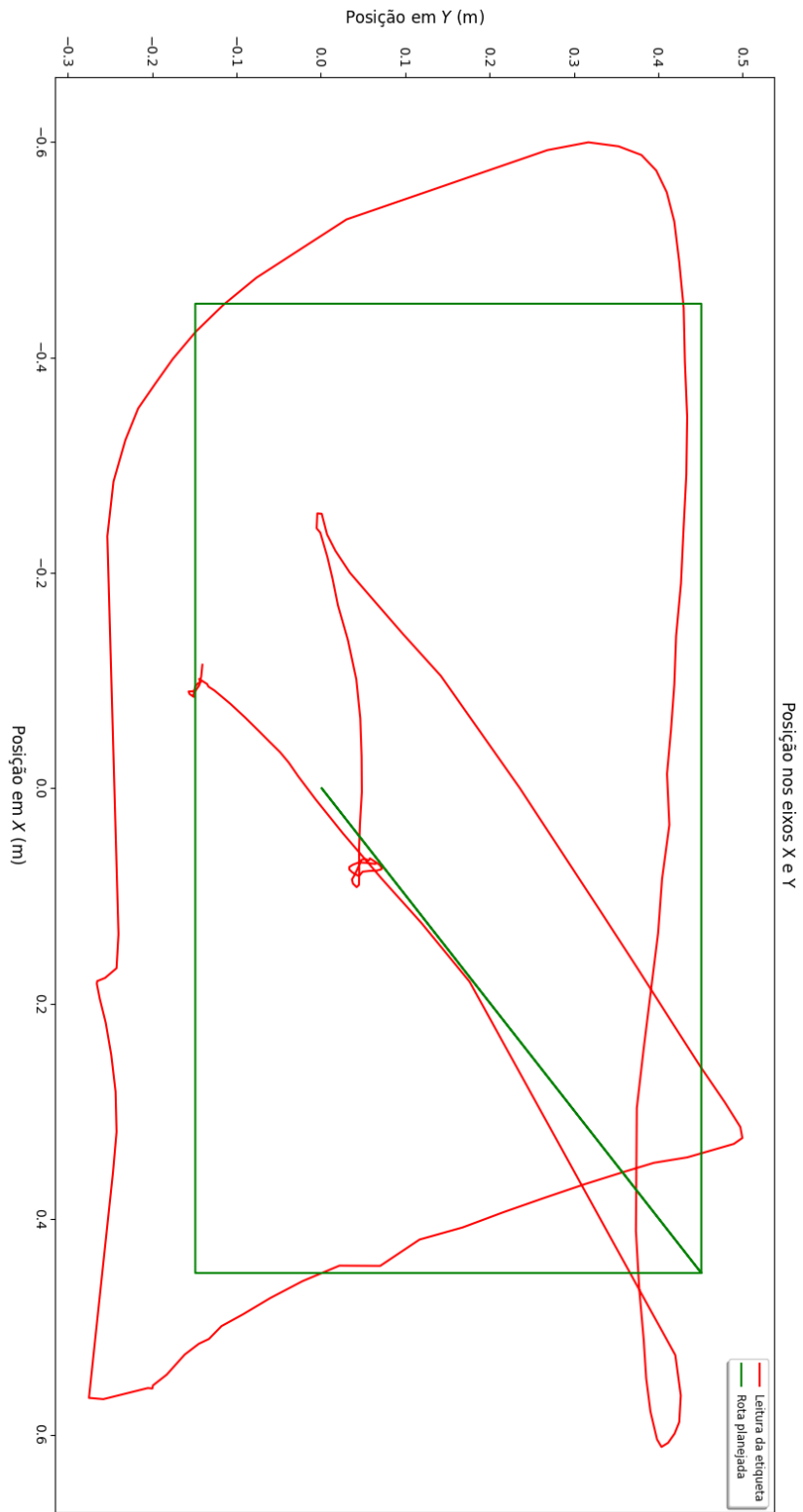
ANEXO K – Rota realizada pelo VANT Figura 25

Rota 3D



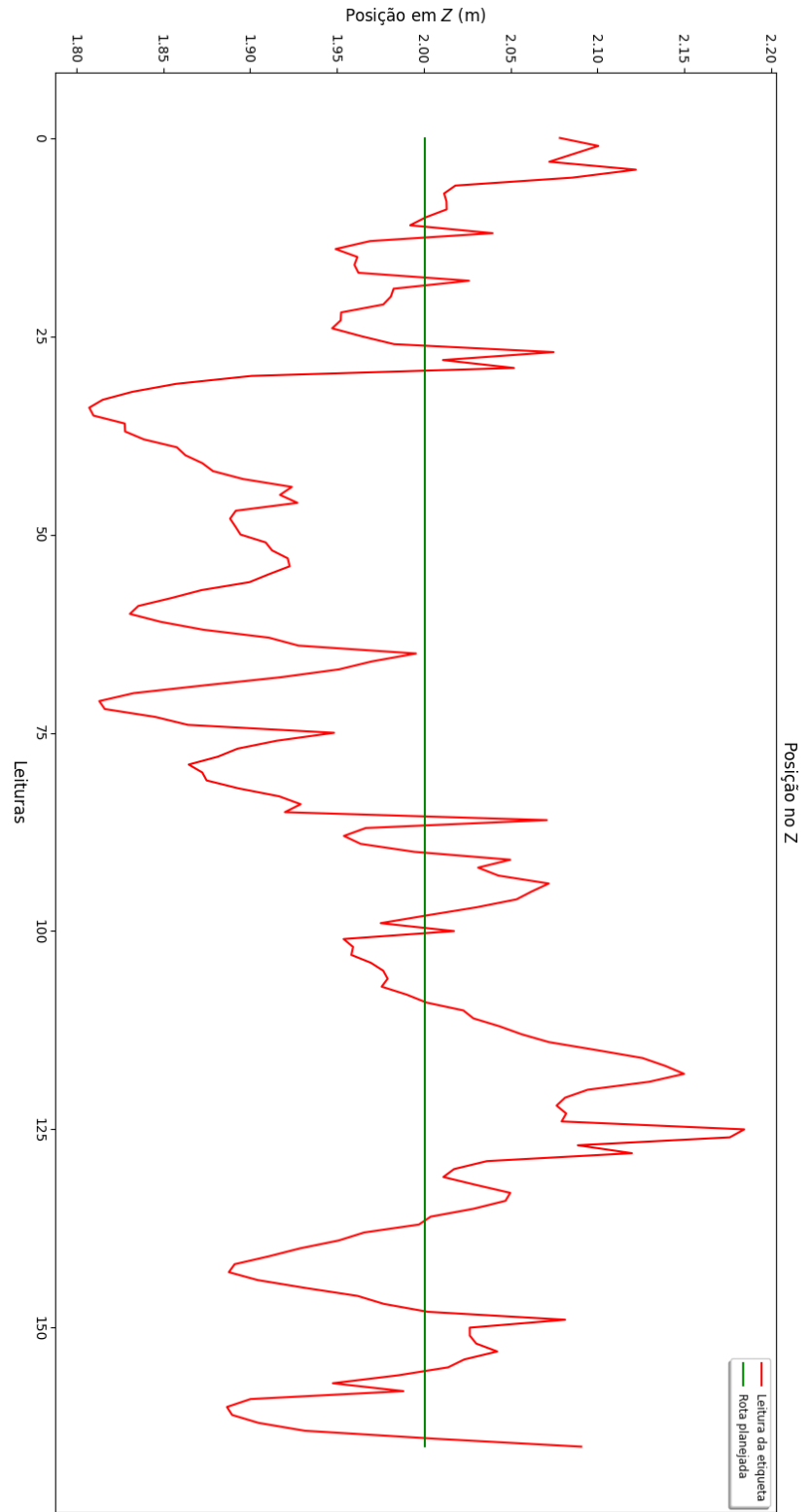
Fonte: Autoria própria.

ANEXO L – Gráficos da posição do VANT Figura 26 (a)



Fonte: Autoria própria.

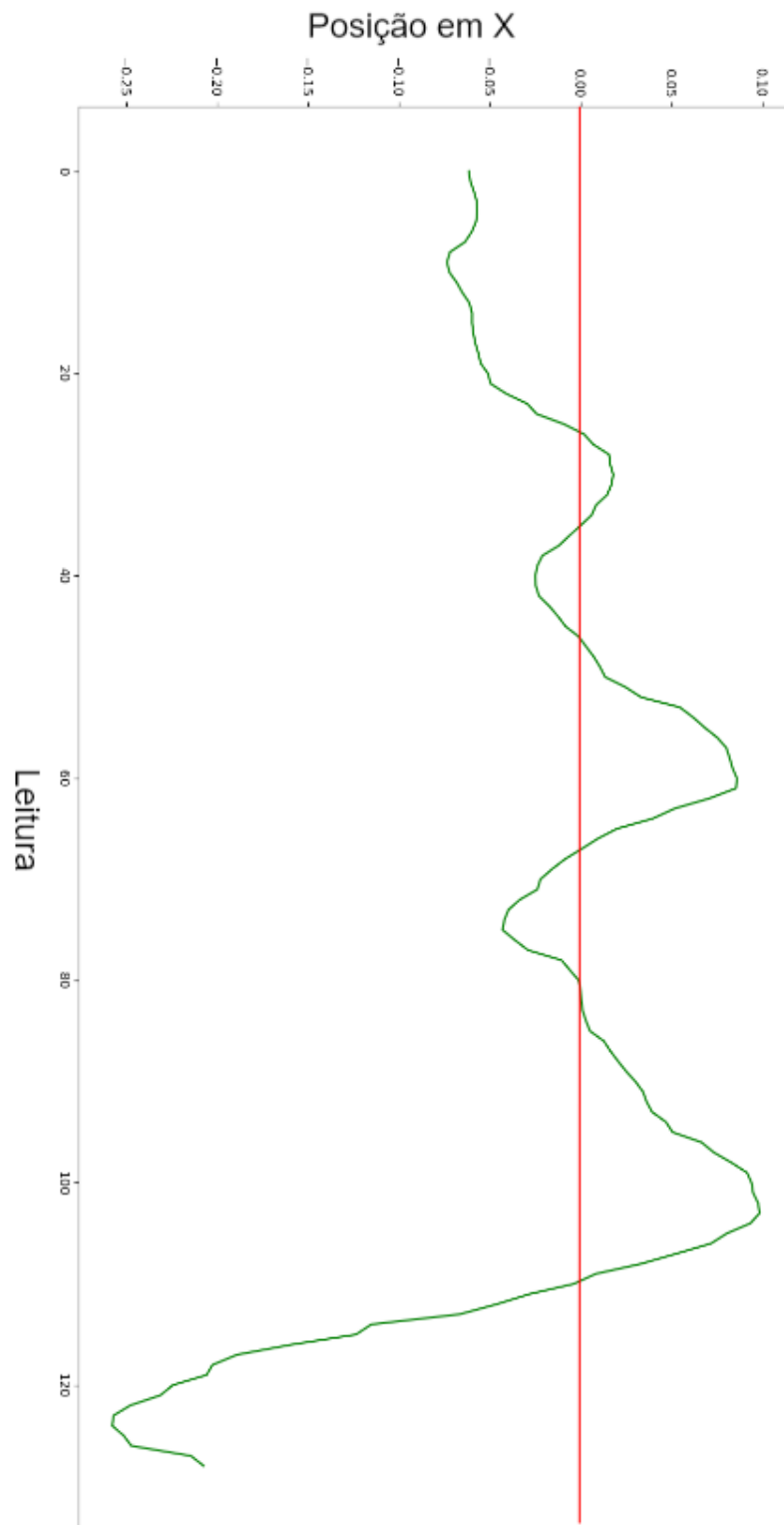
ANEXO M – Gráficos da posição do VANT Figura 26 (b)



Fonte: Autoria própria.

ANEXO N – Gráficos da posição do VANT escravo em relação ao mestre

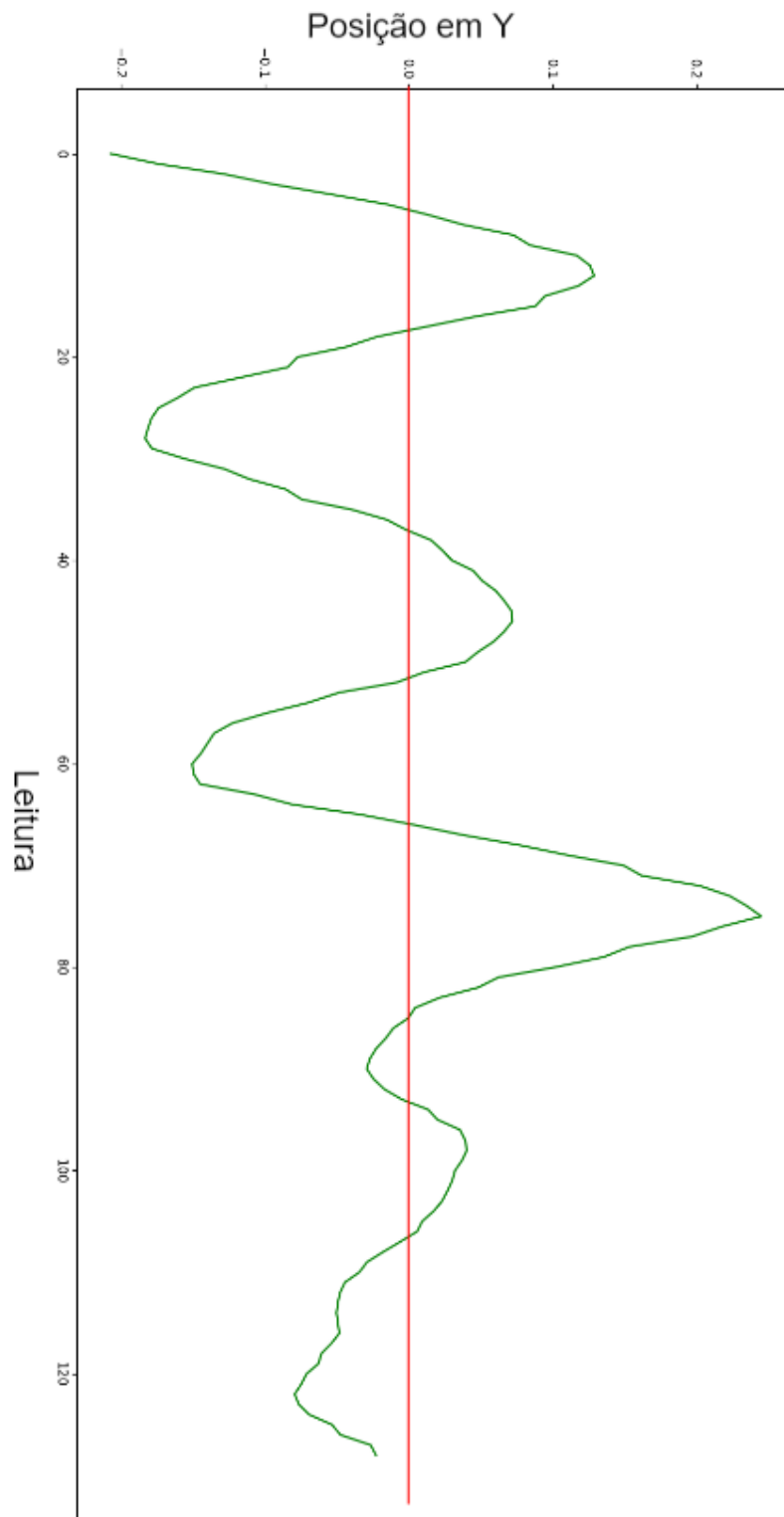
Figura 28 (a)



Fonte: Autoria própria.

ANEXO O – Gráficos da posição do VANT escravo em relação ao mestre

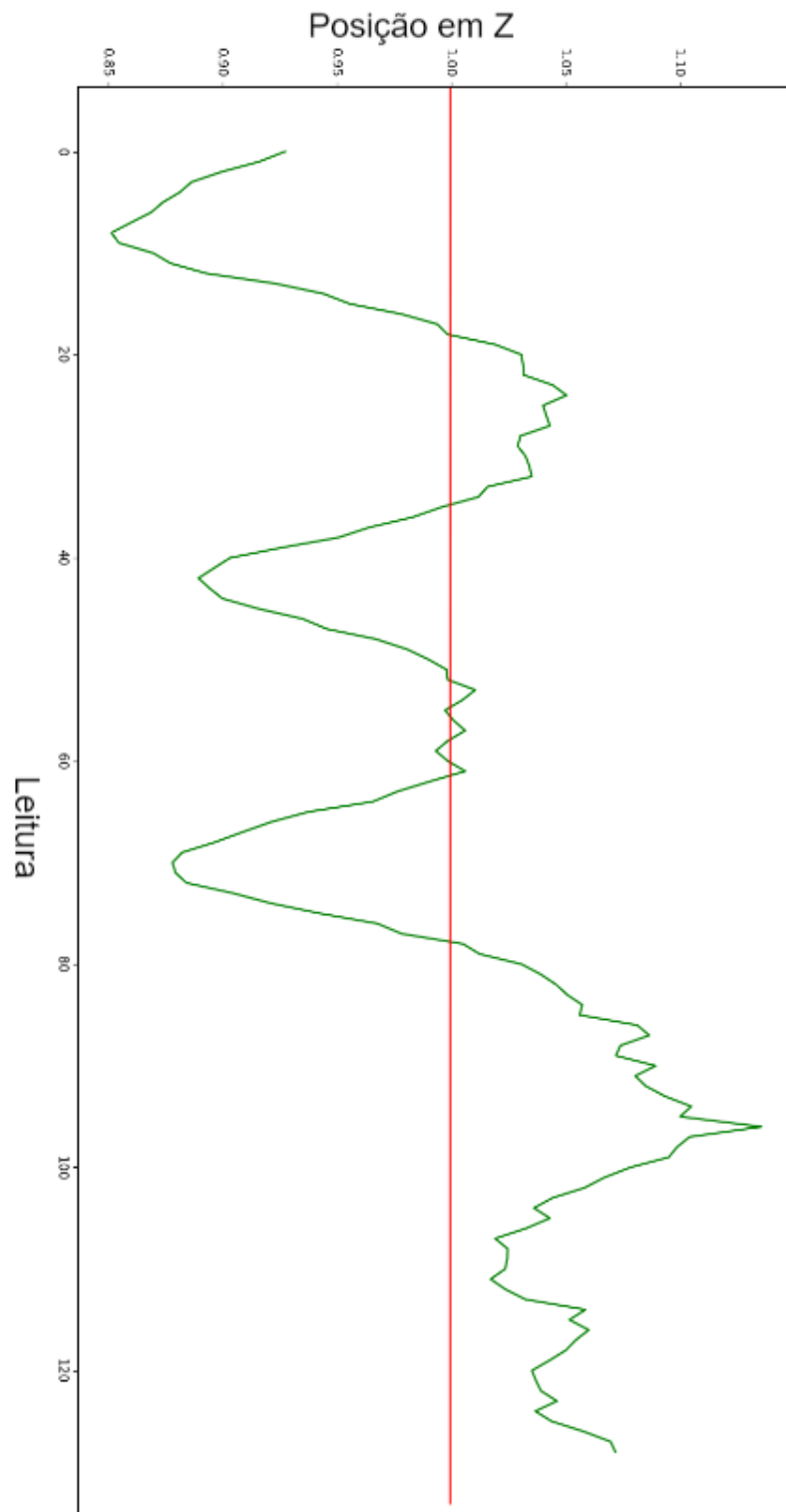
Figura 28 (b)



Fonte: Autoria própria.

ANEXO P – Gráficos da posição do VANT escravo em relação ao mestre

Figura 28 (c)



Fonte: Autoria própria.