

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
FACULDADE DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA
ELÉTRICA

Guilherme Augusto Namorato Pussente

Controle Adaptativo de Alinhamento e Distância de um VANT em
Missões de Inspeção de Linhas de Distribuição

Juiz de Fora

2026

Guilherme Augusto Namorato Pussente

**Controle Adaptativo de Alinhamento e Distância de um VANT em
Missões de Inspeção de Linhas de Distribuição**

Tese apresentada ao Programa de Pós-graduação em Engenharia Elétrica da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do título de Doutor em Engenharia Elétrica. Área de concentração: Sistemas Eletrônicos

Orientador: Prof. Eduardo Pestana de Aguiar, D.Sc.

Coorientador: Prof. André Luís Marques Marcato, D.Sc.

Juiz de Fora

2026

Guilherme Augusto Namorato Pussente

Controle Adaptativo de Alinhamento e Distância de um VANT em Missões de Inspeção de Linhas de Distribuição

Tese apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do título de Doutor em Engenharia Elétrica. Área de concentração: Sistemas Eletrônicos

Aprovada em 04 de março de 2026.

BANCA EXAMINADORA

Prof. Dr. Eduardo Pestana de Aguiar - Orientador
Universidade Federal de Juiz de Fora

Prof. Dr. André Luis Marques Marcato - Coorientador
Universidade Federal de Juiz de Fora

Prof. Dr. Alexandre Bessa dos Santos
Universidade Federal de Juiz de Fora

Prof. Dr. Manuel Arturo Rendón Maldonado
Universidade Federal de Juiz de Fora

Profa. Dra. Marley Maria Bernardes Rebuzzi Vellasco
Pontifícia Universidade Católica do Rio de Janeiro

Prof. Dr. André Gustavo Scolari Conceição
Universidade Federal da Bahia

Juiz de Fora, 23/01/2026.



Documento assinado eletronicamente por **Eduardo Pestana de Aguiar, Professor(a)**, em 04/03/2026, às 17:55, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Alexandre Bessa dos Santos, Professor(a)**, em 04/03/2026, às 18:21, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Marley Maria Bernardes Rebuzzi Vellasco, Usuário Externo**, em 04/03/2026, às 19:17, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Andre Gustavo Scolari Conceicao, Usuário Externo**, em 04/03/2026, às 19:25, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Andre Luis Marques Marcato, Professor(a)**, em 04/03/2026, às 19:32, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Manuel Arturo Rendon Maldonado, Professor(a)**, em 04/03/2026, às 19:50, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



A autenticidade deste documento pode ser conferida no Portal do SEI-Ufjf (www2.ufjf.br/SEI) através do ícone Conferência de Documentos, informando o código verificador **2838286** e o código CRC **84F48F76**.

RESUMO

A etapa de distribuição de energia elétrica é fundamental para o fornecimento de eletricidade e, conseqüentemente, para o desenvolvimento socioeconômico de uma região. A inspeção das linhas de distribuição constitui uma etapa crucial para a manutenção da qualidade e da continuidade do serviço prestado. Atualmente, essa atividade é realizada por trabalhadores que operam em contato direto com os cabos ou por meio de helicópteros que voam próximos a estruturas energizadas, o que representa riscos significativos à segurança. Com o avanço dos veículos aéreos não tripulados (VANTs ou drones), esses sistemas passaram a ser explorados para inspeções, surgindo, assim, a necessidade de tecnologias de apoio à análise e à tomada de decisão rápida e precisa. Nesse contexto, o presente trabalho propõe uma metodologia para inspeção semiassistida de linhas de transmissão utilizando VANTs, contemplando as tarefas de alinhamento lateral e angular e de seguimento assistido, realizadas por meio de um controle baseado em informações visuais obtidas por uma câmera alinhada ao eixo de simetria do VANT. A identificação das linhas elétricas é realizada com o detector YOLOv8, por meio de um modelo adaptado para a segmentação de objetos. Os pixels correspondentes aos cabos são processados por um algoritmo de regressão linear, que determina os parâmetros do trajeto à frente do VANT e possibilita a estimativa dos erros de posição, utilizados no cálculo das ações de controle. Essas ações podem ser interrompidas a qualquer momento pelo operador, caso seja necessário inspecionar um ponto específico. Para o controle do sistema, foi implementado um controlador híbrido que emprega lógica fuzzy do tipo 2 para ajustar dinamicamente os ganhos do controlador PID, de acordo com a magnitude e o comportamento do sistema. A metodologia foi avaliada em ambiente simulado (Gazebo), demonstrando desempenho superior em relação a controladores de ganho fixo e a abordagens baseadas em lógica fuzzy do tipo 1. Por fim, experimentos em ambiente real comprovaram que o YOLOv8 identifica as linhas com desempenho compatível com aplicações em tempo real, e que a metodologia proposta foi capaz de manter o VANT alinhado aos cabos de energia durante o teste prático.

Palavras-chave: VANT. Segmentação com redes neurais. Lógica fuzzy do tipo 2.

ABSTRACT

The power distribution stage is fundamental to the supply of electricity and, consequently, to the socioeconomic development of a region. The inspection of distribution lines constitutes a crucial step for maintaining the quality and continuity of the provided service. Currently, this activity is carried out by workers who operate in direct contact with the cables or by means of helicopters flying close to energized structures, which poses significant safety risks.

With the advancement of unmanned aerial vehicles (UAVs or drones), these systems have begun to be explored for inspection tasks, giving rise to the need for technologies that support rapid and accurate analysis and decision-making. In this context, the present work proposes a methodology for semi-assisted inspection of transmission lines using UAVs, encompassing the tasks of lateral and angular alignment and assisted following, performed through a control approach based on visual information obtained from a camera aligned with the UAV's symmetry axis.

The identification of power lines is performed using the YOLOv8 detector through a model adapted for object segmentation. The pixels corresponding to the cables are processed by a linear regression algorithm, which determines the parameters of the path ahead of the UAV and enables the estimation of position errors used in the calculation of control actions. These actions can be interrupted at any time by the operator if it is necessary to inspect a specific point.

For system control, a hybrid controller was implemented that employs type-2 fuzzy logic to dynamically adjust the PID controller gains according to the magnitude and behavior of the system. The methodology was evaluated in a simulated environment (Gazebo), demonstrating superior performance compared to fixed-gain controllers and approaches based on type-1 fuzzy logic. Finally, experiments in a real environment confirmed that YOLOv8 identifies the lines with performance compatible with real-time applications, and that the proposed methodology was able to keep the UAV aligned with the power cables during the practical test.

Keywords: UAV. Segmentation with neural networks. Type 2 fuzzy logic.

LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplo de detecções de linhas de energia através de técnicas baseadas em gradientes.	28
Figura 2 – Exemplo de detecções de falhas mecânicas em linhas de distribuição.	30
Figura 3 – Representação de uma CNN simplificada	33
Figura 4 – Arquitetura da rede YOLOv8. Adaptado de [RangeKing 2023].	36
Figura 5 – Núvem de pontos em uma inspeção através de sensor LiDAR	40
Figura 6 – Funções de pertinência de sistemas fuzzy do tipo 1 e do tipo 2.	47
Figura 7 – Tipos de missão abordados neste trabalho	59
Figura 8 – Tela da missão	61
Figura 9 – Esquemático dos passos referentes a metodologia proposta	62
Figura 10 – Exemplo de anotações de imagens em cenários contendo linhas de distribuição	65
Figura 11 – Exemplo de imagem sintética obtida através de uma imagem do <i>dataset</i> original.	66
Figura 12 – Exemplo de inferência através da YOLOv8-seg	67
Figura 13 – Exemplo de inferência dos elementos do sistema elétrico com YOLOv8	69
Figura 14 – Exemplo de máscara binária em cenário com mudança de orientação das linhas.	70
Figura 15 – Passos para obtenção dos parâmetros da trajetória para abordagem IBVS.	73
Figura 16 – Visão aérea do problema abordado neste trabalho	74
Figura 17 – Visão lateral do problema abordado neste trabalho	75
Figura 18 – Processo de calibração da câmera para correção de perspectiva da imagem.	77
Figura 19 – Exemplo da aplicação da Homografia na imagem do drone.	79
Figura 20 – Estimativa dos erros de posição e de alinhamento com base na abordagem IBVS.	80
Figura 21 – Extração de parâmetros visuais através de uma imagem .	81

Figura 22 – Representação dos eixos do drone utilizado neste trabalho	83
Figura 23 – Representação dos eixos do drone utilizado neste trabalho	85
Figura 24 – Representação do bloco IT2_PID utilizado neste trabalho.	86
Figura 25 – Funções de pertinência para o bloco $IT2_PID_{z_e}$.	88
Figura 26 – Funções de pertinência para o bloco $IT2_PID_{\varphi_e}$.	89
Figura 27 – Cenário criado no Gazebo para realização dos testes em ambiente simulado.	93
Figura 28 – Comparativo da magnitude do erro lateral para os quatro controladores abordados	95
Figura 29 – Comparativo do erro lateral acumulado para os quatro controladores abordados.	97
Figura 30 – Comparativo do erro lateral médio para diferentes conjuntos de regras <i>fuzzy</i> .	98
Figura 31 – Comparativo da magnitude do erro angular para os quatro controladores abordados	100
Figura 32 – Comparativo do erro angular acumulado para os quatro controladores abordados.	101
Figura 33 – Comportamento do erro lateral durante missão de seguimento das linhas de distribuição em cenário simulado.	102
Figura 34 – Comportamento do erro angular durante missão de seguimento das linhas de distribuição em cenário simulado.	103
Figura 35 – Comportamento do erro lateral acumulado durante missão de seguimento das linhas de distribuição em cenário simulado.	104
Figura 36 – Comportamento do erro angular acumulado durante missão de seguimento das linhas de distribuição em cenário simulado.	105
Figura 37 – Comparação das métricas de performance para o treinamento da rede YOLOv8-seg	108
Figura 38 – Comparativo de perda de qualidade de imagem do Parrot Bebop 2. Acima, o sinal recebido pelo computador via rede; abaixo, a captura direta da memória do drone.	110
Figura 39 – Detecção dos elementos na linha de distribuição na missão de seguimento	112

Figura 40 – Ponto de partida para os testes de alinhamento em um ambiente real.	114
Figura 41 – Evolução temporal do erro para a missão de alinhamento lateral.	115
Figura 42 – Evolução temporal dos ganhos do controlador IT2_PID para a missão de alinhamento lateral.	116
Figura 43 – Evolução temporal do erro para a missão de alinhamento angular	117
Figura 44 – Evolução temporal dos ganhos do controlador IT2_PID para a missão de alinhamento angular.	118
Figura 45 – Trajetória para o Teste de Seguimento - Cenário 1	119
Figura 46 – Erro lateral para teste de seguimento - Cenário 1	120
Figura 47 – Erro angular para teste de seguimento - Cenário 1	121
Figura 48 – Teste de seguimento em ambiente real - Cenário 1	122
Figura 49 – Trajetória para o Teste de Seguimento - Cenário 2	123
Figura 50 – Erro lateral para teste de seguimento - Cenário 2	123
Figura 51 – Erro angular para teste de seguimento - Cenário 2	124
Figura 52 – Teste de seguimento em ambiente real - Cenário 2	125
Figura 53 – Trajetória para o Teste de Inspeção	126
Figura 54 – Exemplos visuais da missão de inspeção das linhas de distribuição	127
Figura 55 – Erro lateral para teste de inspeção	128
Figura 56 – Erro angular para teste de inspeção	129

LISTA DE TABELAS

Tabela 1 – Versões da rede YOLOv8-seg	37
Tabela 2 – Trabalhos correlatos	55
Tabela 3 – Conjunto de regras <i>fuzzy</i> para adaptação dos ganhos do bloco PID.	87
Tabela 4 – Resultados quantitativos para o controle lateral em cenário simulado	95
Tabela 5 – Novo conjunto de regras proposto.	98
Tabela 6 – Resultados quantitativos para o controle lateral em cenário simulado.	99
Tabela 7 – Resultados quantitativos para o controle angular em cenário simulado	100
Tabela 8 – Resultados quantitativos para a missão de seguimento das linhas em cenário simulado	104
Tabela 9 – Métricas de Performance para modelos YOLO-seg para tarefa de segmentação das linhas de distribuição	107
Tabela 10 – Tempo necessário para inferência da YOLOv8-seg na máquina utilizada nos testes de campo.	109
Tabela 11 – Métricas de Performance para modelo YOLOv8 para detecção de elementos do sistema de distribuição	111
Tabela 12 – Métricas de performance do controlador IT2_PID para alinhamento lateral.	113
Tabela 13 – Métricas de performance do controlador IT2_PID para alinhamento angular.	116

LISTA DE ABREVIATURAS E SIGLAS

ABC	<i>Artificial Bee Colony Algorithm</i>
ACO	<i>Ant Colony Algorithm</i>
CIOU	<i>Binary Cross-Entropy</i>
CIOU	<i>Complete Intersection over Union</i>
DFL	<i>Distribution Focal Loss</i>
CNN	<i>Convolutional Neural Networks</i>
CBS	<i>Circle Based Search</i>
CSA	<i>Crow Search Algorithm</i>
FCN	<i>Fully Convolutional Network</i>
FOU	<i>Footprint of Uncertainty</i>
GA	<i>Genetic Algorithm</i>
GPS	<i>Global Positioning System</i>
GA	<i>Gray Wolf Optimization</i>
HGS	<i>Hunger Games Search</i>
IA	<i>Inteligência Artificial</i>
IAE	<i>Integral of Absolute Error</i>
IBVS	<i>Image-based visual servoing</i>
IMU	<i>Inertial Measurement Unit</i>
LMF	<i>Lower Membership Function</i>
LSD	<i>Line Segment Detector</i>
MCP	<i>Modelo de controle preditivo</i>
MPA	<i>Marine Predators Algorithm</i>
OLS	<i>Ordinary Least Squares</i>
PBVS	<i>Position-based Visual Servoing</i>
PD	<i>Proporcional Derivativo</i>
PID	<i>Proporcional Integrativo Derivativo</i>
PSO	<i>Particle swarm optimization</i>
PBVS	<i>Position-based visual servoing</i>
PCNF	<i>Pulse Coupled Neural Filter</i>
R-CNN	<i>Region Based Convolutional Neural Networks</i>
R-FCN	<i>Region-based Fully Convolutional Networks</i>
ReLU	<i>Rectified Linear Unit</i>
ROS	<i>Robot Operating System</i>

RN	Redes Neurais
SMA	<i>Slime Mould Algorithm</i>
SMC	<i>Sliding Mode Controller</i>
SPPF	<i>Spatial Pyramid Pooling Fast</i>
SSD	<i>Single Shot multibox Detector</i>
UMF	<i>Upper Membership Function</i>
VANT	Veículo Aéreo Não Tripulado
WOA	<i>Whale Optimization Algorithm</i>
YOLO	<i>You Only Look Once</i>

LISTA DE SÍMBOLOS

v_x	Velocidade de arfagem
v_y	Velocidade de rolagem
K_p	Ganho proporcional
K_i	Ganho integral
K_d	Ganho derivativo
K_u	Ganho final
T_u	Período de oscilação
ω_y	Velocidade de guinada
x	Ponto longitudinal da função de estimação trajetória
y	Ponto latitudinal da função de estimação trajetória
Z_e	Erro lateral da trajetória
φ_e	Erro angular da trajetória
l	Distância entre a projeção do ponto central do VANT no plano do objeto e a origem do sistema de coordenadas
P_0	Ponto central do objeto
h	altitude relativa do VANT com relação às linhas
θ_c	Inclinação do eixo ortogonal da imagem e o eixo vertical do VANT
d	distância da câmera e o ponto P_0
x_{ref}	Referência longitudinal para o cálculo dos erros visuais
y_{ref}	Referência latitudinal para o cálculo dos erros visuais
ϕ_{ref}	Referência angular para o cálculo dos erros visuais
Δ_x	Diferença lateral entre a posição real do objeto e a referência adotada
k	Fator de conversão de <i>pixels</i> para metros
f_L	Distância focal da câmera
FOV_H	Campo de visão horizontal da câmera
M_p	Pico máximo
T_p	Tempo de pico
T_s	Tempo de estabilização
T_s	Erro de estado estacionário
P	Precisão
R	Revocação
α	Fator para cálculo de K_i

SUMÁRIO

1	Introdução e Motivação	16
1.1	Objetivo	20
1.2	Contribuições	21
1.3	Publicações Relacionadas	22
1.4	Organização da Tese	23
2	Referencial Teórico	24
2.1	Detecção dos cabos elétricos	24
2.1.1	Métodos baseados em gradientes	25
2.1.2	Métodos baseados em Aprendizado de máquina	27
2.1.2.1	YOLOv8	33
2.1.3	Outros métodos de detecção	38
2.2	Técnicas de Controle	41
2.2.1	Controlador PID	42
2.2.1.1	Controladores PID sintonizados por lógica fuzzy	43
2.2.1.2	Controladores PID sintonizados por redes neurais	50
2.3	Trabalhos Correlatos	51
2.3.1	Conclusões Parciais	54
3	Metodologia Proposta	57
3.1	Detecção das linhas de distribuição	62
3.2	Detecção dos elementos do sistema de distribuição de energia	67
3.3	Estimativa dos parâmetros da trajetória	69
3.4	Cálculo das variáveis de erros visuais	73
3.5	Controle de movimentação	82
3.5.1	Controle adaptativo IT2_PID	85
3.6	Conclusões Parciais	91
4	Resultados e Discussões	92
4.1	Resultados em ambiente simulado	92
4.1.1	Controlador Lateral	94
4.1.2	Efeito da modificação do conjunto de regras	97
4.1.3	Controlador Angular	99

4.1.4	Controle IT2_PID em missão de seguimento das linhas de distribuição - Cenário simulado.	101
4.2	Resultados em ambiente real	105
4.2.1	Análise do modelo YOLOv8 para identificação das linhas de distribuição.	106
4.2.2	Análise do modelo YOLOv8 para a detecção dos elementos da rede de distribuição	109
4.2.3	Testes de controle em ambiente real	112
4.2.3.1	Testes de alinhamento lateral	113
4.2.3.2	Testes de alinhamento angular	115
4.2.3.3	Testes de seguimento das linhas de distribuição	117
4.2.3.3.1	Cenário 1	118
4.2.3.3.2	Cenário 2	122
4.2.3.4	Testes de inspeção das linhas de distribuição	125
4.2.4	Discussões	129
5	Conclusões e Propostas Futuras	132
5.1	Conclusões	132
5.1.1	Propostas Futuras	134
	REFERÊNCIAS	136

1 Introdução e Motivação

A inspeção das linhas de distribuição é um processo de suma importância para a manutenção da qualidade da energia fornecida a uma região. Eventuais falhas no sistema elétrico podem levar a paralisações ou desperdícios que acarretam prejuízos financeiros, além de potenciais perdas humanas e danos ambientais. O Brasil perde 20.28% da energia produzida na distribuição no sistema elétrico, sendo essas perdas oriundas de questões técnicas geradas por características elétricas dos equipamentos e condutores do sistema, além de perdas causadas por fatores comerciais, como problemas na medição e fraudes na distribuição. Este valor de perda é bastante elevado se comparado a outros países vizinhos, como Argentina (9.9%), Peru (9.3%), Colômbia (11.3%) e Chile (5.6%) [Silva et al. 2012]. Esses dados evidenciam a necessidade de melhorias no processo de inspeção da rede elétrica e, sendo assim, uma possível saída é o uso de sistemas robóticos que auxiliem na detecção de anomalias, permitindo tomadas de decisões mais precisas e assertivas.

Ao longo dos anos, o processo de identificação de falhas e manutenção das linhas de energia é realizado majoritariamente por operadores humanos que ficam em contato direto com elementos de alta tensão, sendo este o método mais utilizado para inspeções da rede elétrica ao redor do mundo ainda nos dias atuais [Alhassan et al. 2020]. Invariavelmente, o uso de trabalho manual nesses casos resulta em acidentes devido à queda do operador ou eletrocussão. No Brasil, acidentes envolvendo o setor elétrico são, em média, 4.8 vezes maiores que outros setores da economia [Silva 2015]. Possíveis alternativas para este problema são a inspeção por meio de helicópteros ou a análise terrestre na qual operadores realizam o trabalho por meio de binóculos. Esses métodos, porém, se mostram ineficientes, lentos e muitas vezes custosos.

Com o advento da robótica, diversas alternativas surgiram para que as inspeções do sistema elétrico se tornem cada vez mais rápidas e seguras. Zengin propôs um robô que se acopla aos cabos e, por meio de sensores, consegue detectar o arqueamento das linhas [Zengin et al. 2020]. Já em [Gao et al. 2020], um robô foi desenvolvido para análise interna dos cabos de energia, no qual imagens de raios

X são utilizadas em conjunto com técnicas de aprendizagem profunda (do inglês, Deep Learning) para detecção de defeitos. Os trabalhos supracitados apresentam soluções nas quais existe o contato direto de componentes eletrônicos com o sistema de distribuição de energia. O principal problema relacionado ao uso desses tipos de robô é a dificuldade de locomoção dos mesmos devido às diferentes configurações de cabos, além da presença dos postes de distribuição, bem como outros elementos como árvores e construções. Além disso, outro ponto a se considerar é a interferência eletromagnética causada pela proximidade dos equipamentos eletrônicos com o sistema de distribuição de energia.

Considerando estas limitações, o emprego de VANTs (Veículos aéreos não-tripulados), uma vertente da robótica que vem recebendo crescente atenção do meio acadêmico nos últimos anos, apresenta-se como uma ferramenta promissora nos processos de inspeção de estruturas nas quais as linhas de distribuição estão inseridas. Estes veículos possuem características importantes que permitem sua empregabilidade em diversas tarefas, onde os trabalhos eram realizados majoritariamente por trabalhadores humanos.

A capacidade de voo por áreas extensas, por exemplo, permite que estes robôs sejam utilizados para o monitoramento de grandes estruturas, como barragens [Zhao et al. 2021, Pinto et al. 2020]. A possibilidade de diferentes modos de voo e a flexibilidade quanto a dimensão permitem que os drones consigam atuar em espaços reduzidos com grande concentração de máquinas e/ou fluxo de pessoas como no âmbito industrial [Mourtzis, Angelopoulos e Panopoulos 2021, Pathak, Mukherjee e Misra 2021]. Outro ponto a se considerar é a capacidade de drones em integrar sensores (GPS, câmeras, LiDAR, etc) com atuadores (braços robóticos, servo motores e pulverizadores), permitindo a sua atuação no âmbito agrícola como pulverizador de plantas [Chen et al. 2021, Wang et al. 2020] e no meio urbano como na entrega de mercadorias [Ko, Park e Ko 2022, Han e Jung 2020].

Todas as potencialidades apresentadas indicam que os drones também podem ser empregados em tarefas envolvendo linhas de distribuição, seja como uma ferramenta de auxílio para manutenção, quanto para detecção de falhas no sistema que podem levar a interrupções na distribuição de energia e acidentes. Desta forma, diversos trabalhos da literatura propõem soluções relacionadas à

utilização de drones para inspeções do sistema elétrico.

Com relação à manutenção das linhas, Uzakov, Nascimento e Saska propuseram um sistema no qual um drone contendo ferramentas deve seguir um operador enquanto o mesmo atua no processo de manutenção da rede elétrica [Uzakov, Nascimento e Saska 2020]. Lopez propôs a aplicação de um drone para identificação e limpeza de isoladores elétricos em linhas de distribuição através de um jato de água acoplado em sua estrutura [Lopez et al. 2021]. Para a identificação de falhas nas linhas, Guan utilizou um sensor LiDAR para detecção dos componentes do sistema de distribuição de energia através de um processo de classificação da nuvem de pontos, e os componentes defeituosos são identificados através de processos de aprendizado de máquina aplicados nas imagens obtidas por uma câmera RGB [Guan et al. 2021]. Já no trabalho de [Ayoub e Schneider-Kamp 2021], a rede YOLOv4-tiny foi utilizada para detecção de possíveis falhas em componentes elétricos em um sistema de distribuição, sendo que o planejamento da trajetória foi feito por controle manual.

Um dos requisitos necessários para um sistema autônomo de inspeção de linhas de distribuição de energia através de drones é a implementação de um sistema de controle robusto que permita que as missões sejam realizadas a uma distância adequada dos cabos, de forma a reduzir a interferência magnética que pode afetar os sensores e atuadores do drone, bem como a comunicação com a estação de solo que gerencia toda a operação. Sistemas de inspeção baseados integralmente em controle manual são pouco eficientes, uma vez que demandam tempo e operadores qualificados em manusear o drone em regiões próximas a elementos de alta tensão que podem danificar todo o aparato de inspeção, além de oferecer riscos às pessoas transeuntes e aos partícipes da operação. Além disso, o controle manual pode afetar a movimentação do drone e, conseqüentemente, a qualidade das imagens, dificultando a identificação de possíveis problemas no sistema.

A grande maioria dos drones comercializados nos dias atuais dispõem de câmeras RGB. Assim, uma alternativa viável para que o posicionamento do veículo seja adequado durante a etapa de inspeção são as metodologias baseadas em algoritmos de controle por meio de informações visuais. Esta classe de controladores pode ser subdividida em duas abordagens principais: PBVS (do Inglês, Position-

based visual servoing) e IBVS (do Inglês, Image-Based visual servoing).

Técnicas de controle baseadas em PBVS utilizam informações visuais para obtenção da reconstrução tridimensional da posição da câmera com relação ao objeto por meio de coordenadas cartesianas. Embora esta abordagem faça com que o sistema seja globalmente estável, algumas premissas devem ser respeitadas, como a determinação precisa dos parâmetros de calibração da câmera e do modelo tridimensional do objeto. Imprecisões nessas medições podem afetar o desempenho geral do sistema e, como o controle é aplicado em termos das coordenadas 3D do ambiente e não do plano da imagem, o objeto pode sair do campo de visão da câmera [Chaumette 2007].

As abordagens baseadas em IBVS utilizam-se de informações extraídas no plano de imagens para determinar as ações de controle a serem tomadas, de forma que o objeto de interesse seja mantido em uma região específica da imagem, geralmente na porção central. Desta forma, evita-se que o objeto saia do campo de visão da câmera, o que é imprescindível no processo de inspeção de estruturas. Um dos problemas desta vertente é a estimação da profundidade da imagem, diferentemente das coordenadas cartesianas da imagem em 2D (*comprimento, altura*). Além disso, como os comandos de controle são determinados para o plano da imagem e não para o plano do robô, alguns movimentos indesejados podem ser aplicados ao robô. Isso pode levar, por exemplo, que o robô se mova para uma direção com a câmera apontando para a direção oposta, gerando a falsa impressão de que o robô está se distanciando do objeto quando, na verdade, ele está se aproximando, podendo causar a perda momentânea do referencial de controle [Corke e Hutchinson 2001]. Para mitigar este efeito, uma estratégia viável é a imposição de restrições nas velocidades aplicadas ao drone, de forma que possíveis erros na estimativa da posição do objeto, bem como movimentos imprevistos não afetem a capacidade de manter a referência no campo de visão da câmera [Shakiba et al. 2023].

As informações acerca das técnicas de controle servo-visuais citadas acima indicam que uma etapa crucial para metodologias de seguimento de objetos é a detecção precisa dos mesmos. Considerando que o drone sobrevoa os cabos de energia, um dos desafios para a identificação e localização dos mesmos por meio de informações extraídas de imagens é a distinção dos elementos de interesse

(cabos, torres, isolantes) com elementos no fundo dos ambientes pelos quais o drone passa, como árvores, asfalto, ferrovias dentre outras construções. Outro ponto a ser considerado são os diferentes níveis de iluminação do ambiente que podem afetar a eficácia das detecções. Estas questões podem fazer com que as linhas não sejam detectadas (falsos negativos) ou que outros componentes, como marcações no asfalto, sejam tratados como linhas (falsos positivos), o que pode afetar o desempenho global do sistema de monitoramento .

Abordagens clássicas utilizam-se de características como linearidade e paralelismo dos cabos de distribuição e empregam algoritmos de detecção de formas geométricas como a Transformada de Hough [Hough 1962] e LSD (do inglês, *Line Segment Detector*) [Gioi et al. 2012]. Embora sejam técnicas que exijam pouco esforço computacional, as mesmas não conseguem distinguir elementos retilíneos diferentes, aumentando a quantidade de falsos positivos. Recentemente, com o advento das técnicas de aprendizado de máquina, algoritmos de detecção de objetos como o YOLO (do inglês, You Look Only Once) [Redmon et al. 2016] e R-CNN (do inglês, Region Based Convolutional Neural Networks) [Girshick 2015] vêm sendo empregados em metodologias de monitoramento do sistema elétrico, uma vez que conseguem lidar melhor com os problemas levantados anteriormente devido aos modelos passarem por uma etapa de treinamento que permite a identificação de objetos em um processo análogo ao processo de aprendizado humano.

1.1 Objetivo

O principal objetivo do presente trabalho é propor uma metodologia para a inspeção de linhas de distribuição de energia por meio de um drone, utilizando técnicas de controle assistido através de uma abordagem IBVS. O sistema como um todo deve ser capaz de identificar e determinar de forma precisa a posição e a orientação das linhas que são enviadas a um controlador, que tem como objetivo manter o drone alinhado e paralelo com as linhas conforme realiza o seguimento. O operador em solo possui um *joystick* capaz de enviar comandos de início e fim da missão, bem como a possibilidade de controlar o drone em caso de alguma anormalidade. No computador que realiza o processamento da operação, é apresentada uma interface gráfica com informações de comandos para operação do

drone e dados de telemetria do mesmo, como o nível de bateria, altitude, velocidade e distância relativa das linhas. Além disso, conforme o drone segue as linhas, é realizada a identificação dos postes de energia e outros elementos como cruzetas e isoladores, sendo que o operador tem liberdade de sair do modo de seguimento autônomo para realizar uma inspeção manual de algum elemento de interesse, com a possibilidade de voltar ao modo de seguimento ou finalização da missão.

A técnica de detecção das linhas deve ser capaz de determinar de forma precisa a posição e orientação das mesmas em diversos cenários de ambiente e iluminação, além de diferenciar os cabos de outros elementos retilíneos nas imagens. Já a técnica de controle deve ser capaz de lidar com incertezas inerentes a sistemas baseados em posicionamento de objetos em imagens, bem como imprecisões nos parâmetros da câmera. Com isso, foi proposta a identificação das linhas através de um detector de objetos baseado em aprendizado de máquina, enquanto as velocidades aplicadas ao drone para a tarefa de alinhamento e seguimento são tratadas por controladores PID sintonizados por lógica *fuzzy*. É importante salientar que o foco deste trabalho está na inspeção de sistemas de distribuição de energia. Apesar disso, a metodologia proposta pode ser adaptada para sistemas de transmissão devido a similaridades morfológicas entre cabos de energia para ambos os cenários, sendo necessária a escolha de um drone adequado com relação a características como interferência eletromagnética, autonomia e qualidade da imagem.

1.2 Contribuições

Esta tese visa apresentar uma metodologia para identificação, seguimento e inspeção de linhas de distribuição por drones. Um modelo de rede neural convolucional é utilizado para a segmentação das linhas e as informações visuais são passadas para um controlador PID sintonizado por lógica fuzzy do tipo 2 de forma a manter o drone posicionado acima das linhas. Além do objetivo principal, este estudo apresenta as seguintes contribuições:

- **Criação e compartilhamento de bancos de imagens contendo linhas de energia e postes de distribuição;**

Para este trabalho, foram construídos dois bancos de imagens: um focado nas linhas de distribuição e transmissão em diversos cenários e o outro contendo postes de distribuição, que permitem estudos acerca do treinamento de diversos modelos de identificação de objetos. Além das imagens reais, foram introduzidas imagens artificiais com variações no brilho e na orientação do objeto, de forma que o sistema de identificação seja capaz de prover detecções precisas, permitindo que o drone realize a tarefa de inspeção sem movimentos bruscos ou o risco de perda do referencial. O banco de imagens das linhas de distribuição e transmissão pode ser acessado por meio do *link*, enquanto o banco de imagens referente aos componentes do sistema de distribuição de energia pode ser acessado por meio do *link*.

- **Identificação das linhas e controle de velocidade em tempo reduzido;**

A identificação das linhas, o cálculo de erros e o controle do drone são realizados em tempo compatível com aplicações que requerem baixa latência, através do processamento das informações por meio de uma máquina em solo e a transmissão dos sinais de controle sem fio por meio de subscrição e publicação de tópicos em ROS (do inglês, *Robot Operating System*).

- **Adaptabilidade da metodologia de controle para diversos modelos de drones.**

A metodologia de controle é facilmente adaptável para diferentes modelos de drone, sem a necessidade do equacionamento das dinâmicas do drone. Além disso o modelo de segmentação é capaz de lidar com diferentes tipos de disposição de cabos de distribuição presentes no meio urbano.

1.3 Publicações Relacionadas

Este trabalho resultou em dois artigos de revista, listados abaixo :

- "UAV Power Line Tracking Control Based on a Type-2 Fuzzy-PID Approach", Pussente, G. A., de Aguiar, E. P., Marcato, A. L., & Pinto, M. F, Robotics. MDPI, 2023. [Pussente et al. 2023].

Este artigo apresenta a metodologia de controle para o alinhamento e seguimento de linhas de distribuição baseado em informações visuais através de um controlador PID sintonizado por um bloco de lógica fuzzy do tipo-2. Os ganhos do controlador são adaptados de acordo com o estado e o comportamento do erro, sujeitos a erros de medições. O controlador proposto foi comparado com controladores de ganhos fixos em simulações realizadas no simulador Gazebo.

- "Advanced drone-based powerline inspection using image segmentation and adaptive visual control", Pussente, G. A. ,Biundini, I. Z. ,de Aguiar, E. P., & Marcato, A. L. Journal of the Brazilian Society of Mechanical Sciences and Engineering , 2025.

Esse artigo apresenta a integração completa do sistema de controle IT2-PID com um sistema de detecção baseado em redes neurais para o alinhamento e seguimento de linhas de distribuição. O detector YOLOv8 foi treinado com um banco de imagens contendo linhas de energia em diversos cenários e o controle do drone foi testado em um cenário contendo linhas de distribuição reais para as tarefas de alinhamento angular,lateral e seguimento autônomo.

1.4 Organização da Tese

Além deste capítulo, que faz uma breve introdução acerca da aplicabilidade dos drones em diversas áreas, a tese de doutorado é dividida em quatro capítulos adicionais. No Capítulo 2 é feita uma revisão bibliográfica com relação aos temas principais contidos na metodologia proposta, destacando-se o ponto no qual o presente trabalho se insere. No Capítulo 3, é apresentada uma descrição minuciosa dos passos necessários para a construção da metodologia proposta. No Capítulo 4 são apresentados os resultados para a identificação das linhas de distribuição, bem como a detecção de elementos presentes no sistema elétrico, além da integração do sistema de detecção com o sistema de controle para a tarefa de alinhamento e seguimento das linhas em cenários artificiais, através do simulador Gazebo e em ambientes reais. Por fim, o Capítulo 5 apresenta as conclusões finais do trabalho, bem como os possíveis passos a serem abordados.

2 Referencial Teórico

Este capítulo apresenta informações necessárias para o entendimento e posterior implementação da metodologia proposta neste trabalho. Inicialmente, a Seção 2.1 trata dos métodos de detecção das linhas de distribuição de energia através de drones. A Seção 2.2 apresenta diferentes abordagens de controle empregadas para drones. A Seção 2.3 faz um pequeno comparativo dos trabalhos relacionados à inspeção de sistemas de distribuição de energia, inserindo as contribuições da presente pesquisa.

2.1 Detecção dos cabos elétricos

Uma das principais etapas no processo de planejamento de trajetória em sistemas robóticos é a identificação de pontos de interesse, tanto para evitar possíveis obstáculos quanto para a identificação e seguimento dos mesmos. No âmbito industrial, por exemplo, um robô pode ser utilizado para o seguimento de uma linha industrial e, para isso, deve ser capaz de identificar pessoas ou máquinas de forma a evitar colisões que gerem acidentes. Assim, tanto a linha quanto os objetos ao redor devem ser mapeados em tempo hábil, de forma que o robô consiga parar caso a linha esteja obstruída.

Embora a localização espacial e a identificação de objetos ou pontos de interesse sejam triviais para os seres humanos, existem diversas questões a serem consideradas com relação à implementação de algoritmos que realizem essa tarefa, auxiliando sistemas autônomos de navegação. Um dos problemas inerentes a sensores como o LiDAR, por exemplo, são medições ruidosas que podem levar a imprecisões na localização dos objetos e, por consequência, afetar o planejamento da trajetória [Le, Cheng e Liu 2023]. Esses sensores também possuem valores elevados, o que inviabiliza seu uso em algumas aplicações [Campbell et al. 2018].

Câmeras óticas, por sua vez, vêm recebendo grande atenção quanto à sua aplicabilidade em sistemas robóticos devido ao seu custo reduzido e a quantidade de informações que podem ser extraídas através das imagens. Nos dias atuais, grande parte dos drones comercializados já possui câmeras integradas em seu sistema. Além disso, ambientes robóticos como o ROS conseguem extrair seus dados de

maneira rápida e prática, permitindo à integração do sistema sensorial com os atuadores responsáveis pela movimentação do robô. Apesar destas características, técnicas de identificação de objetos baseadas em imagem possuem algumas questões importantes a serem levadas em conta.

Imagens de câmeras são compostas por *pixels*, pequenas grades que contêm informações como cores e intensidade luminosa em uma determinada coordenada da imagem. Estes valores, porém, podem ser afetados por fatores como diferentes níveis de iluminação, sobreposição de objetos e ruídos na imagem causados por falhas na lente ou na própria transmissão do fluxo de imagens entre o drone e a máquina responsável pelo processamento das informações. Por estes motivos, os sistemas robóticos baseados em informações visuais geralmente empregam um pré-processamento das imagens, que são técnicas que visam suprimir informações redundantes e destacar características importantes no processo de identificação dos objetos. Muitas técnicas também empregam filtros após as detecções, que reduzem a quantidade de falsos positivos [Maharana, Mondal e Nemade 2022].

As técnicas empregadas para a detecção dos cabos de energia podem ser divididas em três grupos principais: os algoritmos baseados em gradientes, técnicas baseadas em aprendizado de máquina e detecção tridimensional através de sensores a laser. Algumas metodologias empregam a junção de duas ou mais técnicas, permitindo detecções mais precisas e confiáveis.

2.1.1 Métodos baseados em gradientes

Muitas técnicas de detecção por imagens fazem um mapeamento vetorial em busca de conjuntos de *pixels* que contenham informações semelhantes, podendo assim ser candidatos a um determinado objeto, de acordo com sua forma ou cor. As linhas de distribuição em imagens aéreas possuem características como linearidade, paralelismo e uniformidade de cor. Tendo conhecimento destas informações, técnicas generalizadas de classificação de objetos baseadas em formas geométricas podem ser utilizadas para selecionar potenciais candidatas a linhas. Uma vez que não existe a distinção desses objetos para outros elementos retilíneos, é necessária uma etapa posterior de filtragem que permita a eliminação de possíveis falsos positivos.

A técnica mais usual para detecção de formas geométricas é a Transformada

de Hough. Ela baseia-se no agrupamento de pontos compreendidos nas bordas dos objetos (do inglês, Edge Points). Para isso, utiliza-se técnicas como o Detector de Bordas de Canny [Canny 1986] como uma etapa de pré-processamento da imagem e, em seguida, os pontos de bordas colineares são agrupados em um processo de votação no qual apenas os grupos mais votados são tratados como linhas. Apesar de ser uma técnica simples, é bastante dependente de um ajuste de parâmetros e realiza uma detecção generalizada, sendo necessário um pós-processamento dos objetos detectados.

No campo de detecção de linhas de distribuição de energia em imagens aéreas, Li propôs uma metodologia de detecção em que um filtro PCNF (*Pulse Coupled Neural Filter*) é utilizado para remoção dos ruídos do ambiente e em seguida empregou uma versão adaptada da Transformada de Hough proposta por [Fernandes e Oliveira 2008] que modificou o processo de votação das candidatas a retas, reduzindo assim a detecção de elementos espúrios [Li et al. 2010]. Em [Tian, Wang e Zhu 2015] foram utilizados filtros morfológicos para o destaque dos cabos em diversas direções na imagem e, em seguida, a detecção foi realizada através da Transformada de Hough com restrições baseadas no paralelismo das linhas. Liu empregou uma versão adaptada da Transformada de Hough Randomizada [Xu, Oja e Kultanen 1990], utilizando-se da premissa de que as linhas de distribuição ficam compreendidas em uma área enclausurada e possuem larguras específicas nas imagens e, por fim, segmentos com o mesmo ângulo são agrupados [Liu, Hou e Ju 2017].

Dentre outras metodologias baseadas em gradientes que foram empregadas na detecção de cabos de energia se destaca a Transformada de Radon [Helgason e Helgason 1999] que aplica uma transformação de domínio referente aos parâmetros das retas, sendo os picos no gráfico deste novo domínio possíveis candidatos a segmentos retilíneos. Em [Chen et al. 2016] esta técnica foi empregada para detecção de cabos de distribuição de energia em imagens de satélite. Já Yan emprega a Transformada de Radon para identificação de pequenos segmentos de reta e em seguida utiliza um Filtro de Kalman para agrupar os segmentos em retas identificadas como cabos de energia [Yan et al. 2007]. Outro detector de linhas encontrado na literatura é o LSD (*Line Segment Detector*), que baseia-se na

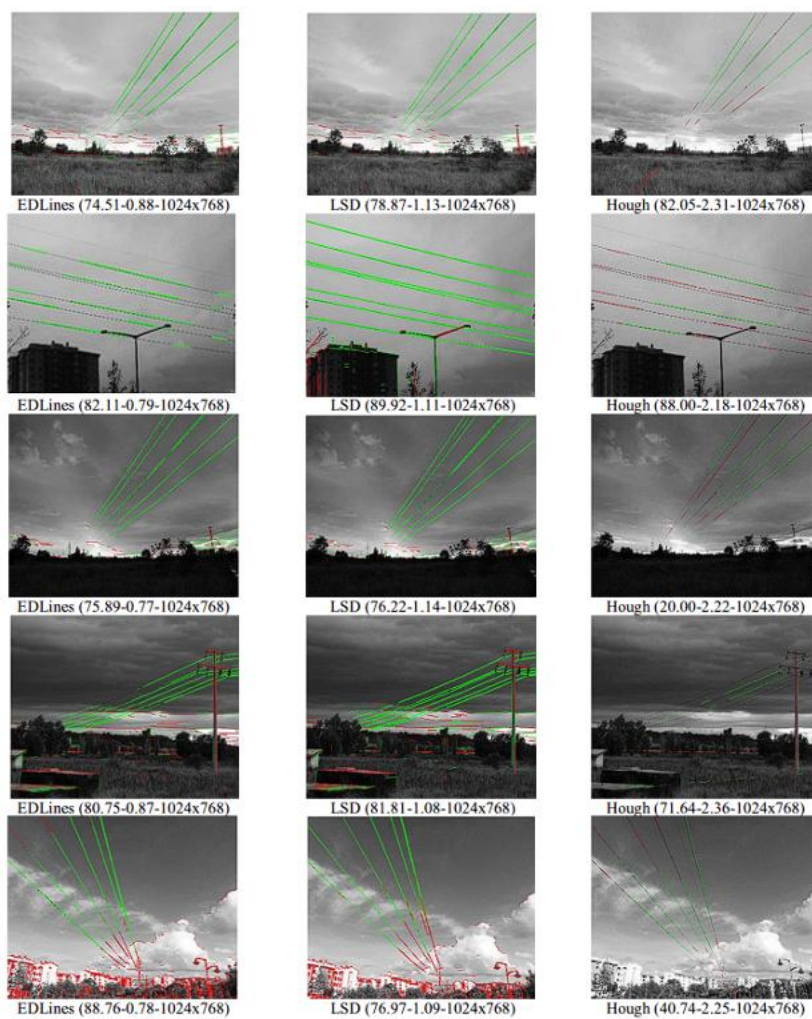
transformação do espaço de cores da imagem para escala de cinza e, em seguida, os segmentos (grupos de pixels em que existe uma mudança abrupta no nível da cor cinza) são agrupados em regiões com angulações semelhantes sendo estas candidatas a linhas. Lu, Yu e Yang propõem uma metodologia baseada no LSD junto com uma Rede Neural Convolutiva para a identificação das linhas [Luo, Yu e Yang 2021]. Já em [Zhao, Dong e Zuo 2022], um modelo de Campo Aleatório de Markov (*Markov Random Field*) utiliza-se das regiões candidatas a linhas de transmissões obtidas pelo LSD para eliminação de falsos positivos baseados no ângulo e distância entre as linhas.

O trabalho de [Yetgin, Senturk e Gerek 2015] propôs a utilização do detector EDLines [Akinlar e Topal 2011] para detecção de linhas e fez um comparativo com a transformada de Hough e LSD. A Figura 1 apresenta exemplo de detecções de linhas de distribuição através de algumas técnicas abordadas nesta seção. As linhas em verde são os segmentos corretamente detectados, enquanto as linhas em vermelho são segmentos que deveriam ser detectados (falsos negativos) ou foram detectados incorretamente (falsos positivos). É possível notar que as técnicas baseadas em gradientes não são capazes de distinguir os cabos de energia de outros elementos retilíneos, sendo necessário, portanto, técnicas adicionais que aplicam um conhecimento prévio das características desses elementos nas imagens para filtrar os resultados, o que não garante uma confiabilidade adequada para um sistema de inspeção autônomo.

2.1.2 Métodos baseados em Aprendizado de máquina

Com o advento da computação na década de 1950, surgiu o termo Inteligência Artificial (IA), que compreende toda atividade na qual uma máquina é capaz de adquirir informações sobre o ambiente ao seu redor e, a partir disso, tomar alguma decisão por conta própria, de forma a otimizar um determinado processo. Com o passar dos anos, o avanço tecnológico permitiu que as máquinas passassem a aprender de acordo com o andamento do processo, buscando alternativas que gerem resultados cada vez mais satisfatórios, analogamente à inteligência humana [Shinde e Shah 2018]. Esse campo específico de IA foi chamado de Aprendizado de Máquina e, conforme os dispositivos eletrônicos vêm sofrendo uma drástica

Figura 1 – Exemplo de detecções de linhas de energia através de técnicas baseadas em gradientes.



Fonte: Retirado de [Yetgin, Senturk e Gerek 2015].

diminuição no tamanho, com um aumento cada vez maior na performance, abriu-se um grande leque de aplicações no campo da robótica, sobretudo para robôs de pequeno porte como os drones.

No campo da agricultura, por exemplo, técnicas baseadas em Aprendizado de máquina vêm sendo empregadas em diversas tarefas, tais como: previsão de rendimento de culturas agrícolas em uma abordagem não invasiva utilizando drones [Senthilnath et al. 2016], detecção de doenças em plantas [Chung et al. 2016], identificação de erva daninha próxima a outras plantas [Pantazi, Moshou e Bravo 2016] e classificação comportamental de gado [Dutta et al. 2015]. No âmbito industrial, técnicas inteligentes podem ser empregadas para o planejamento de trajetórias de robôs em uma linha de produção ou para movimentação precisa de um braço robótico respeitando restrições cinemáticas e físicas [Soori, Arezoo e Dastres 2023]. Salhaoui propôs um sistema de monitoramento e controle de uma fábrica de concreto utilizando-se de um drone conectado a um sistema de computação em nuvem que gera todo o sistema de produção [Salhaoui et al. 2019].

Com relação a inspeções do sistema elétrico, diversas aplicações envolvendo aprendizado de máquina vêm sendo exploradas, permitindo a identificação de elementos como torres, isolantes e os cabos de energia, além da detecção de inconsistências no sistema. Com relação às falhas elétricas, medidores de tensão e corrente elétrica podem passar por técnicas de classificação baseadas em séries temporais [Nawaz et al. 2024] ou modelos de aprendizado baseados em regras [Gutierrez-Rojas et al. 2022] para identificação de possíveis falhas no sistema.

No âmbito das falhas mecânicas, imagens aéreas obtidas por drones podem passar por algoritmos de detecção de objetos, permitindo a extração de elementos defeituosos, agilizando o processo de manutenção do sistema de distribuição. Adou, Xu e Chen propuseram o uso da rede YOLOv3 para detecção de isoladores elétricos e eventuais falhas mecânicas nos mesmos [Adou, Xu e Chen 2019]. Já o trabalho de [Jenssen, Roverso et al. 2019] propôs uma metodologia mais generalizada baseada na rede *Single Shot multibox Detector* (SSD) para detecção de diversos componentes, além de possíveis falhas mecânicas como cruzetas e postes quebrados e/ou apodrecidos. O trabalho de [Li et al. 2020] utiliza uma rede EpNet para detecção de ninhos de pássaros em linhas de distribuição. Por fim, o trabalho de

determinada operação e transmite seu resultado para a camada seguinte com a ponderação dos pesos que podem aprimorar ou denegrir a saída. Este processo é chamado de *feed-forward* e, como as CNNs possuem diversas camadas, essas redes fazem parte da classe de Aprendizado Profundo (do inglês, *Deep Learning*).

CNNs possuem três tipos de camadas principais. As diferentes metodologias baseadas neste tipo agrupam diversas dessas camadas, visando à melhoria do modelo em detectar padrões de diferentes formas e tamanhos. Essas camadas podem ser resumidas da seguinte forma [O'Shea e Nash 2015] :

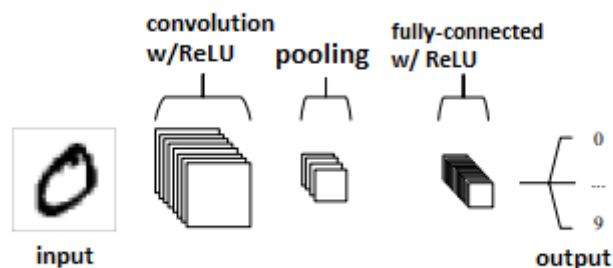
- **Camadas Convolucionais:** Esta bloco é o principal elemento de uma CNN. Uma imagem de entrada, ao entrar nesta camada, é percorrida por meio de *kernels*, filtros capazes de detectar determinadas características em uma região da imagem, processo que recebe o nome de convolução. O resultado desta operação é chamado de mapa de ativação, que são regiões relevantes da imagem para detecção de determinada classe (um padrão ou um objeto, por exemplo). O processo de aprendizado nesse tipo de camada é realizado por uma função de ativação, que atua no sentido de introduzir não-linearidades ao modelo, permitindo que a rede reconheça padrões complexos entre os dados de uma forma mais satisfatória. Existem diversos tipos de função de ativação, sendo a Unidade Linear Retificadora (ReLU, do inglês, Rectified Linear Unit) a mais utilizada devido a sua capacidade de ativar apenas um grupo de neurônios, não denegrindo a capacidade da rede em detectar outros padrões. Em camadas iniciais, os filtros tendem a detectar características simples, como bordas e contrastes. À medida que a profundidade da rede aumenta, os filtros passam a identificar estruturas mais complexas, como partes de objetos ou padrões completos.
- **Camada de *Pooling*:** A função principal desta camada é reduzir a dimensionalidade das camadas e, por consequência, a quantidade de parâmetros da rede. Esse processo pode ser interpretado como uma forma de resumir a informação presente em pequenas regiões da imagem, preservando as características mais relevantes enquanto reduz o tamanho da representação. Além de diminuir o custo computacional, essa operação também contribui

para tornar o modelo menos sensível a pequenas variações na posição dos objetos na imagem. Nesta etapa, os mapas de ativação obtidos nas camadas convolucionais sofrem uma redução através de uma funções *maxpool* que computam o valor máximo dentro de uma região de dados e funções *averagepool* que calculam a média dentro da região. Estes processos percorrem a imagem em sua totalidade permitindo que a informação contida em dois ou mais *pixels* seja transcrita em somente uma. Devido a natureza destrutiva destes processos, geralmente são escolhidos *kernels* de tamanho 2 para essas operações.

- **Camadas *Fully connected*:** Por fim, n neurônios correspondentes a n classes a serem detectadas pela rede são conectados aos neurônios da camada anterior nesta etapa. Nessa camada, são inferidas pontuações que indicam a probabilidade de determinado padrão estar presente na imagem. Além disso, podem ser inseridas funções de ativação, como a ReLU, de forma a melhorar a expressividade e performance das detecções ou classificações. Enquanto as camadas convolucionais e de *pooling* atuam principalmente na extração de características visuais da imagem, as camadas *fully connected* são responsáveis por interpretar essas características e realizar a decisão final da rede. Dessa forma, elas funcionam de maneira semelhante a um classificador tradicional que recebe como entrada um conjunto de atributos previamente extraídos.

A Figura 3 apresenta a arquitetura simplificada de uma CNN. Neste exemplo, uma imagem contendo um número escrito manualmente passa pelas camadas da rede que, por fim, classifica o conteúdo em uma saída de 0 a 9.

Figura 3 – Representação de uma CNN simplificada



Fonte: Adaptado de [O'Shea e Nash 2015].

2.1.2.1 YOLOv8

O YOLO, acrônimo para *You Look Only Once*, é um algoritmo de detecção de objetos inicialmente proposto por Redmond et al. [Redmon et al. 2016]. Como o nome sugere, a técnica está inserida na classe de detectores de disparo único, no qual as predições são realizadas através de uma única passagem por toda a rede, diferentemente de detectores de múltiplos disparos como a R-FCN [Dai et al. 2016] e R-CNN [Girshick et al. 2014], nas quais a seleção das regiões candidatas a terem um determinado objeto e a classificação dessas regiões com o posterior refinamento das predições são determinadas em diferentes estágios. Detectores de disparo único possuem maior simplicidade computacional, o que os tornam candidatos para aplicações que requeiram detecções em tempo compatível com aplicações com baixa latência, ao custo de detecções menos precisas, se comparadas a de detectores de múltiplos estágios [Terven e Cordova-Esparza 2023].

Essas características atraíram o interesse do meio acadêmico, que passou a estudar modificações para a estrutura original do YOLO, permitindo detecções mais robustas mantendo um esforço computacional reduzido. Isso fez com que essa variante de detectores passasse a ser explorada em diversas aplicações, sobretudo no campo da robótica em que sistemas baseados em informações visuais devem ser capazes de identificar objetos em cenários complexos em um tempo reduzido, permitindo que o robô realize múltiplas tarefas simultaneamente [Mohammed et al. 2022].

A oitava versão do YOLO, a YOLOv8, foi desenvolvida pela Ultralytics em janeiro de 2023. Sua estrutura, assim como a da sua antecessora direta, a YOLOv5 utiliza o Pytorch [Ketkar et al. 2021], um *framework* com diversos algoritmos focados em aprendizado de máquina na linguagem de programação Python, substituindo a Darknet que, embora eficiente, possui diversas configurações complexas que reduzem a sua aplicabilidade em diversos campos de pesquisa [Hussain 2023]. Vale salientar que, apesar de terem sido lançadas versões mais recentes como a YOLOv11, o YOLOv8 ainda é a versão mais estável da linha de modelos lançados pela Ultralytics até o presente momento, sendo este o motivo de sua utilização neste trabalho.

A estrutura do YOLOv8 é subdividida em três partes principais: a espinha dorsal (*backbone*), o pescoço (*neck*) e a cabeça (*head*). O funcionamento de cada uma dessas etapas é apresentado a seguir.

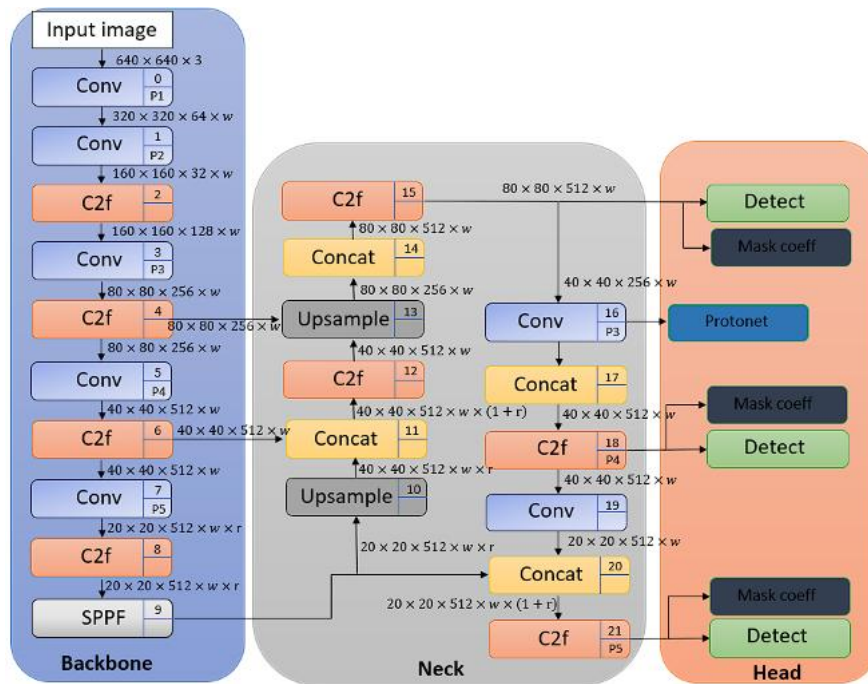
- **Espinha dorsal:** A espinha dorsal é responsável pela extração dos *features* (características relevantes para classificação de uma região de pixels em um objeto) nas imagens. Este processo é realizado pelos blocos de convolução. Para o YOLOv8 são realizadas 5 etapas de convoluções na espinha dorsal, o que permite que a rede detecte *features* de variados tamanhos e tipos de informação detectados. Cada bloco de convolução (Conv) recebe uma entrada e exibe a saída como um mapa de *features* representado por diversas camadas. Estes mapas passam por blocos C2f (*Coarse-to-fine*) que tem como objetivo separar os *features* de diferentes camadas e realizar um processo de concatenação, permitindo que a rede detecte tanto padrões de alto nível (partes do objeto) quanto de baixo nível (bordas ou texturas). Por fim, as informações passam pelo bloco SPPF (*Spatial Pyramid Pooling Fast*), que é uma camada que fixa a dimensionalidade do mapa de *features* através do agrupamento de *features* de diferentes dimensões em um único mapa.
- **Pescoço:** Essa parte da rede funciona como a ligação entre a espinha dorsal e a cabeça. Internamente, é realizado o aprimoramento do mapa de *features* de diferentes tamanhos através da junção de blocos de aumento de dimensão de *features* (*upsample*) com blocos C2f. O objetivo dessas operações é melhorar a

resolução e, conseqüentemente, permitir que a rede consiga distinguir objetos de diferentes tamanhos e características.

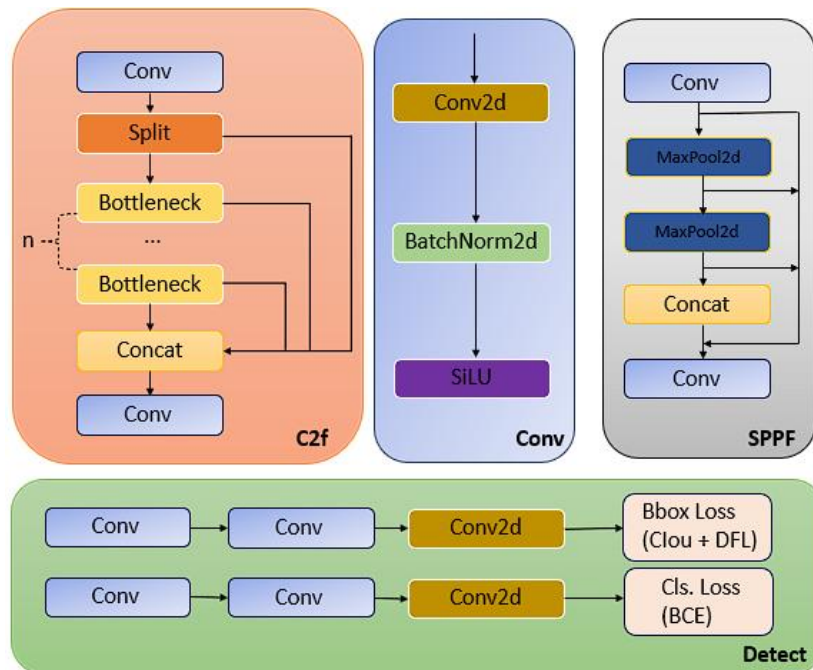
- **Cabeça:** A cabeça da rede tem como objetivo receber os *features* obtidos pelas camadas anteriores e determinar predições sobre as regiões da imagem compreendidas por um determinado objeto por meio do bloco Detect e um valor de pontuação que indica o grau de probabilidade que uma determinada região seja, de fato, um objeto através do bloco Mask Coeff. O bloco Detect é responsável por transformar os mapas de características extraídos pela rede em predições espaciais, estimando a posição, dimensão e localização dos objetos na imagem por meio de *bounding boxes*. Para isso, cada posição do mapa de características é analisada e utilizada para prever coordenadas que delimitam possíveis objetos, juntamente com uma pontuação de confiança associada à presença desses objetos.

A estrutura interna do bloco de detecção (Detect) foi projetada de maneira desacoplada, sendo que uma parte determina a posição e pontuação das *bounding boxes* através das funções de perda CIOU (*Complete Intersection over Union*) e DFL (*Distribution Focal Loss*), e a outra realiza o processo de classificação do objeto por meio da técnica BCE (*Binary Cross-Entropy*) [Wang et al. 2023].

A rede YOLOv8 é utilizada, neste trabalho, para a tarefa de segmentação da imagem. Isso significa que, além das *bounding boxes*, a rede também exibe o objeto segmentado como saída. Para essa tarefa específica, o modelo recebe o nome de YOLOv8-seg, e sua estrutura é similar à versão padrão da rede, com a adição do bloco Protonet, uma FCN (*Fully Convolutional Network*) responsável pelo cálculo da máscara de segmentação de saída e um bloco adicional que computa os coeficientes das máscaras. A Figura 4 apresenta a estrutura principal da YOLOv8-seg, na qual uma imagem entra pela espinha dorsal da rede com a resolução 640x640x3 (comprimento, largura e canais de cores), passa por blocos que reduzem a imagem, facilitando a detecção de padrões do objeto e, por fim, o resultado sai através dos blocos da cabeça. Por questão de organização, cada bloco principal foi apresentado, internamente, como blocos complementares.



(a) Estrutura principal da rede



(b) Estrutura dos blocos complementares

Figura 4 – Arquitetura da rede YOLOv8. Adaptado de [RangeKing 2023].

Uma das questões a serem analisadas ao escolher uma técnica de detecção baseada em redes neurais é a relação entre o esforço computacional e a precisão da detecção. Nesse contexto, a YOLOv8 foi projetada para disponibilizar cinco variantes (*nano*, *small*, *medium*, *large* e *extra-large*), cujas arquiteturas são definidas a partir do ajuste dos parâmetros d (profundidade), h (altura) e r (razão), previamente estabelecidos pelos autores do modelo. Esses parâmetros, apresentados na Tabela 1, determinam a capacidade representacional e o custo computacional de cada variante. Na prática, a escolha da versão a ser utilizada é realizada por meio de um parâmetro de entrada na linha de código referente ao treinamento ou à inferência da rede.

Tabela 1 – Versões da rede YOLOv8-seg

Model	Parâmetro		
	w	h	r
yolov8n-seg	0.33	0.25	2.0
yolov8s-seg	0.33	0.50	2.0
yolov8m-seg	0.67	0.75	1.5
yolov8l-seg	1.00	1.00	1.00
yolov8x-seg	1.00	1.25	1.00

Existem duas abordagens principais para identificação de cabos de energia por meio de técnicas de aprendizado de máquina: técnicas baseadas em detecção de objetos e técnicas baseadas em segmentação de imagem.

A primeira classe de detectores baseia-se na determinação de *bounding boxes*, regiões retangulares que compreendem todo o objeto. Desta forma, a imagem entra na rede e a saída são as regiões de uma determinada classe com uma pontuação correspondente. O trabalho de [Son et al. 2023] propõe a utilização da rede YOLOv5 para a detecção de diversos segmentos das linhas, sendo cada segmento tratado como um único objeto. Yuan et al. apresentaram a rede YOLOv5 para identificação das linhas de distribuição e elementos externos presentes que podem gerar falhas no sistema [Yuan et al. 2023]. Vemula et al. [Vemula e Frye 2020] propõem a utilização da rede YOLACT para detecção da linha primária de distribuição, bem como outros elementos da rede elétrica. Uma desvantagem desta vertente de detectores para a metodologia proposta neste trabalho é que, como o resultado do processo é

uma região retangular na qual o objeto está inserido, seria necessária a utilização de outras técnicas citadas anteriormente para a determinação da orientação dos cabos de transmissão, que enfrentam problemas como a não diferenciação de cabos com outros elementos retilíneos.

Já as metodologias baseadas em segmentação de imagem têm como objetivo identificar uma máscara de segmentação, que compreende os cabos de distribuição, diferentemente da abordagem de *bounding boxes*, na qual a região detectada é sempre retangular. Uma vantagem dessa vertente é a possibilidade de determinar tanto a posição dos cabos quanto a orientação das mesmas, sendo essa uma característica de interesse para sistemas que busquem a inspeção autônoma das linhas. Uma possível desvantagem da utilização da segmentação em relação a modelos de detecção é o esforço computacional aumentado, tendo em vista que, para estes modelos, são adicionadas camadas extras responsáveis pela extração de todos os *pixels* compreendidos pelo objeto. Nessa vertente, Zhang et al. [Zhang et al. 2019] propõem uma rede CNN baseada na arquitetura VGG16 com algumas modificações nas camadas *Fully connected*, permitindo que as linhas sejam segmentadas em imagens contendo ambientes complexos. Já o trabalho de [Stambler, Sherwin e Rowe 2019] utiliza-se de CNNs para reconstrução tridimensional dos cabos nas imagens. Por fim, Surav et al. [Saurav et al. 2019] propõem o uso de banco de imagens contendo imagens visíveis e de espectro infravermelho de cabos de energia para o treinamento da rede *Nested U-Net* que realiza a segmentação das linhas na imagem.

2.1.3 Outros métodos de detecção

Além das técnicas apresentadas anteriormente, que se baseiam na utilização de câmeras, uma outra vertente que pode ser utilizada para a determinação da localização dos cabos de energia são sistemas baseados em identificação a laser, nos quais uma fonte emissora de radiação é utilizada sobre a região a ser mapeada ou por meio da determinação prévia da posição dos postes de distribuição através de dados de GPS, do inglês, *Global Positional System*, com a possibilidade de integração com uma técnica baseada em visão computacional para o refinamento da movimentação entre as torres.

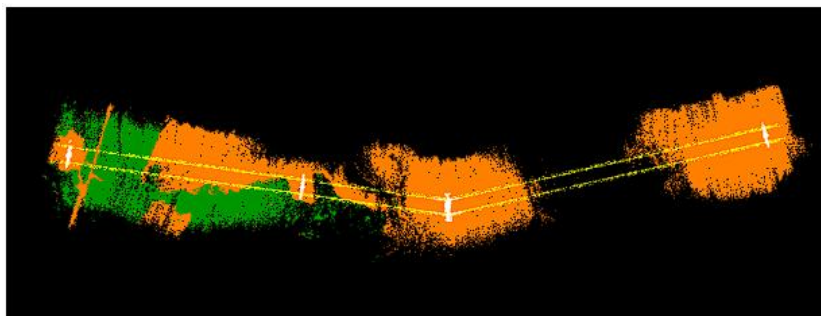
As metodologias baseadas em identificação a laser funcionam através da

passagem de uma fonte de radiação em um material sólido e, pela frequência do sinal refratado, é possível inferir a distância do objeto em relação ao emissor das ondas. Assim, pode se obter um mapa tridimensional da área ao redor do sensor, composto por uma nuvem de pontos contendo informação de coordenadas e distância do objeto. Este tipo de tecnologia é conhecido como LiDAR, que é bastante utilizada para inspeção de estruturas, mapeamento topográfico, veículos autônomos, entre outros [Qin et al. 2018].

O trabalho de [Teng et al. 2017] propôs a utilização de um sensor do tipo LiDAR para o mapeamento de linhas de energia por meio de um drone, e um exemplo de nuvem de pontos obtida pelo sensor pode ser visto na Figura 5. É possível observar duas seções retilíneas em amarelo, referentes às linhas, além de outros elementos, como as torres em branco e pontos em verde e amarelo referentes ao solo. Esse mapa pode ser utilizado em conjunto com a identificação por imagem, permitindo um sistema robusto, mesmo na falta de iluminação ou na falha de um dos sensores. Além disso, os sensores a laser são capazes de detectar outros obstáculos, permitindo que o drone os evite. As desvantagens deste tipo de técnica são o custo dos sensores se comparados às câmeras e o peso adicional que será adicionado ao drone.

Outros trabalhos na literatura também exploram o uso de sensores LiDAR como ferramenta de localização das linhas, em conjunto com outras técnicas capazes de filtrar a nuvem de pontos em busca de padrões que permitam a identificação das linhas. Em [Sohn, Jwa e Kim 2012], o mapa de pontos é passado para um Campo Aleatório de Markov, de modo que as linhas sejam classificadas e os padrões não retilíneos sejam processados por outros classificadores, a fim de identificar esses elementos. Já em [Azevedo et al. 2019], o mapa de pontos é dividido em pequenas grades e os conjuntos de pontos que apresentam propriedades colineares são agrupados e classificados como linhas de distribuição. Wang et al. [Wang et al. 2018] aplica a Transformada de Hough no mapa de pontos para determinar os pontos compreendidos pelos cabos.

Figura 5 – Nuvem de pontos em uma inspeção através de sensor LiDAR



Fonte: Retirado de [Teng et al. 2017].

Com relação às metodologias de inspeção baseadas no conhecimento prévio da posição das torres, [Zhou et al. 2016] utiliza dados de GPS para o posicionamento inicial da missão e, em seguida, realiza a detecção das linhas por meio da aplicação do Detector de Bordas de Canny e da Transformada de Hough em uma região de interesse, determinada pelos *frames* anteriores. Em [Park et al. 2020], as inspeções são realizadas exclusivamente por meio de dados de GPS inseridos no planejador de voo interno do drone. Já em [He, Zeng e Hu 2019], o planejamento de rotas do drone é inicializado de forma manual e, posteriormente, ajustado por leituras de GPS. Em seguida, o modelo tridimensional das torres/linhas é determinado por um sensor LiDAR. As desvantagens da utilização de dados de GPS para a inspeção de linhas de distribuição são a falta de cobertura em algumas localidades e a imprecisão das leituras de coordenadas, que podem levar à perda do referencial a ser observado.

O presente trabalho utiliza uma técnica baseada em aprendizado de máquina para a identificação das linhas durante o processo de inspeção sem o conhecimento prévio da posição das torres e dos cabos. A escolha dessa vertente se deu devido à presença de outros elementos retilíneos próximos às linhas, que podem ser tratados como falsos positivos por técnicas de identificação genéricas, podendo afetar o desempenho da malha de controle do drone durante a missão. Desta forma, foi escolhida a rede YOLOv8 adaptada para a tarefa de segmentação. A região compreendida entre as linhas é tratada como um único objeto, e a saída do processo

é uma máscara de pontos que faz parte desse objeto. Essas informações são passadas a um controlador que visa manter o objeto em uma posição específica da imagem durante a inspeção.

2.2 Técnicas de Controle

Outro ponto a ser considerado para a implementação de uma metodologia de inspeção de linhas de distribuição através de drones é a introdução de um nó de controle capaz de manter o objeto a ser observado na região delimitada da imagem, independentemente de a missão exigir inspeções estáticas, nas quais o drone deve mover-se apenas no plano horizontal, ou em cenários em que deve avançar por uma porção das linhas que podem apresentar mudanças repentinas em sua orientação. Além disso, questões como imprecisões na estimativa da posição e fatores externos, como o vento, devem ser levados em conta no projeto.

A teoria de controle fornece o arcabouço necessário para a modelagem, análise e síntese de controladores aplicados a sistemas dinâmicos, permitindo avaliar propriedades como estabilidade, factibilidade e desempenho sob diferentes condições de operação [Ogata 2003].

Um sistema de controle pode ser descrito como um mecanismo que ajusta as entradas de um sistema dinâmico com o objetivo de regular suas variáveis de saída de acordo com referências previamente definidas [Nise 2020].

Os primeiros sistemas de controle utilizavam dispositivos mecânicos para medir o estado da planta e promover ações de controle por meio dos atuadores. Com o passar das décadas, tanto os sensores quanto os atuadores se tornaram mais precisos, permitindo ajustes de controle com uma resolução cada vez menor e uma frequência de atuação cada vez maior. Além disso, com o advento da microeletrônica, os componentes necessários para o controle de sistemas reduziram de tamanho e de preço, facilitando a automação de sistemas robóticos, nos quais os drones estão inseridos.

Sistemas de controle de drones baseados em IBVS utilizam informações provenientes de câmeras ópticas e outros sensores para a determinação de sinais de controle que são traduzidos em níveis de corrente que devem ser aplicados a

cada rotor para que um determinado movimento seja executado. O bloco principal que recebe os sinais de entrada e calcula a saída para a planta é chamado de controlador. Na literatura, existem diversos tipos de controladores para as mais variadas aplicações. Esta seção traz uma breve introdução das metodologias empregadas para sistemas robóticos, mais especificamente aqueles utilizados para o controle de drones, apresentando as vantagens e desvantagens de cada vertente, além da técnica empregada neste trabalho.

2.2.1 Controlador PID

O controlador proporcional–integral–derivativo (PID) é amplamente utilizado no controle de sistemas dinâmicos devido à sua estrutura simples e à sua eficácia em uma ampla gama de aplicações [Borase et al. 2021]. Sua implementação requer baixo esforço computacional, o que o torna particularmente adequado para sistemas embarcados com recursos limitados, como veículos aéreos não tripulados.

O funcionamento desta classe de controlador está na passagem de um sinal de erro, o qual está relacionado à diferença entre o estado atual e o estado desejado do sistema, em três blocos de controle. O sinal de saída introduzido na planta é o somatório das três parcelas do controlador, que são ponderadas por meio de ganhos escolhidos de acordo com a performance desejada do sistema.

O controlador PID, em sua forma tradicional, continua a ser utilizado em muitas aplicações devido às qualidades mencionadas anteriormente. Entretanto, a sua aplicabilidade sofre restrições quanto ao controle de sistemas não lineares sujeitos a incertezas de medição, como é o caso dos drones [Oersted e Ma 2023]. Outro problema recorrente no que tange ao uso de controladores PID é a escolha dos ganhos que serão aplicados. Abordagens clássicas, como o Método de Ziegler-Nichols, o Método de Cohen-Coon e o Método de ganho e fase, são amplamente utilizadas. Embora essas técnicas necessitem que a função de transferência do sistema seja bem definida, o que pode ser uma questão não trivial para sistemas robóticos complexos [Ogata 2003].

Pensando nessas questões, algumas adaptações foram propostas ao longo do tempo, permitindo que metodologias baseadas no controle PID pudessem ser empregadas em sistemas multisensoriais, como drones. Essas modificações

geralmente introduzem blocos adicionais às metodologias clássicas, permitindo que, por exemplo, os ganhos do controlador se adaptem durante o processo, de forma que o robô tenha comportamentos distintos dependendo da magnitude ou variação das variáveis de erro, o que não é possível em abordagens de ganhos fixos [Lopez-Sanchez e Moreno-Valenzuela 2023]. Com isso, ao longo dos anos, diversos autores propuseram técnicas auxiliares para a sintonização adaptativa dos ganhos do controlador PID, como a lógica *fuzzy* e redes neurais. As subseções a seguir apresentam algumas melhorias propostas para controladores PID, bem como suas aplicações no controle de drones.

2.2.1.1 Controladores PID sintonizados por lógica fuzzy

A computação moderna foi projetada com base na teoria booleana, na qual toda a informação é representada através da lógica binária por meio de dois estados: 0 ou 1. Dessa forma, de acordo com a lógica binária, perguntas triviais como "O robô está perto do alvo ?" ou "Está frio hoje ?" só poderiam ser respondidas por meio de "sim ou não". Essa abordagem levanta alguns questionamentos, como: "Qual a definição de perto ?". Para uma pessoa, a resposta pode ser 10 centímetros, enquanto para outra, 1 metro. Noções como 'frio' ou 'quente' também são relativas. Esses exemplos mostram como as incertezas estão presentes no nosso dia a dia e, por consequência, os problemas de engenharia também precisam lidar com essas incertezas. Sensores e atuadores são dispositivos eletromecânicos suscetíveis a imprecisões e, além disso, fatores externos também exercem influência no comportamento dos processos. Com isso, existem questionamentos sobre como inserir as incertezas nas tomadas de decisão, uma vez que abordagens determinísticas são rígidas quanto a classificação de eventos.

Pensando nessas limitações da lógica clássica, Zadeh propôs a Teoria da Lógica *Fuzzy* [Zadeh 1965]. Nessa abordagem foi introduzido o conceito de conjuntos *fuzzy*, nos quais o pertencimento de um elemento a uma determinada classe não é restrito a valores absolutos. Em vez disso, cada elemento pode pertencer a um conjunto com diferentes graus de pertinência, representados por valores contínuos no intervalo $[0, 1]$.

Esses graus de pertinência são definidos por funções de pertinência que

descrevem o quanto um determinado elemento satisfaz a um conceito linguístico. Por exemplo, ao avaliar a temperatura ambiente, em vez de classificá-la de forma rígida como “fria” ou “quente”, a lógica *fuzzy* permite representações intermediárias, como “pouco frio”, “relativamente frio”, “frio”, “relativamente quente” ou “muito quente”, sendo que cada uma dessas categorias pode possuir um determinado grau de pertinência no intervalo unitário.

Dessa forma, a lógica *fuzzy* fundamenta-se na teoria dos conjuntos *fuzzy*, permitindo representar incertezas e transições graduais entre classes. Isso contrasta com a lógica clássica baseada em conjuntos nítidos (*crisp sets*), na qual o pertencimento a uma determinada classe ocorre de forma estritamente binária, assumindo apenas os valores 0 (não pertence) ou 1 (pertence) [Kosko e Isaka 1993].

Sistemas baseados em lógica *fuzzy* são compostos por quatro etapas principais que podem ser resumidas da seguinte forma [Ross 2005]:

- **Fuzzificação:** Considerando um problema de engenharia, inicialmente, as variáveis de entrada, também chamadas de antecedentes, estão em sua forma numérica exata (do inglês, *crisp numbers*) ao serem obtidas por meio de um sensor digital. Para o funcionamento da lógica *fuzzy*, esses valores exatos precisam ser processados por funções de pertinência, que determinam seu grau de pertencimento ao conjunto *fuzzy*, podendo ser transcritos em variáveis linguísticas (temperatura baixa, média e alta, por exemplo). Desta forma, uma determinada temperatura pode pertencer a um ou mais conjuntos, sendo que o grau de pertinência é ditado pelas funções de pertinência. A passagem de valores numéricos para conjuntos *fuzzy* é chamada de Fuzzificação.
- **Base de regras:** As regras *fuzzy* expressam relações lógicas entre os antecedentes e consequentes (as variáveis de saída) através de operações do tipo SE-ENTÃO que são conhecidas como implicações. O conjunto de regras adotadas em um sistema é geralmente construído de acordo com o conhecimento prévio do especialista e traduz o comportamento do processo de maneira simples e eficaz, uma vez que é expresso por meio de declarações linguísticas.
- **Inferência:** A etapa de inferência recebe os conjuntos *fuzzy*, definidos pelas funções de pertinência, e o conjunto de regras *fuzzy* para determinar a saída do

sistema, que, inicialmente, também é representada por conjuntos *fuzzy* para inferências de Mamdani [Mamdani e Assilian 1975]. No início desta etapa, os graus de pertinência associados às variáveis que compõem o antecedente de cada regra são avaliados. Quando o antecedente é formado por mais de uma condição linguística, os respectivos graus de pertinência são combinados por meio de operadores fuzzy, como os operadores *AND* ou *OR*, resultando em um único grau de ativação para a regra considerada. Em seguida, cada regra passa pelo processo de implicação, que aplica um operador *fuzzy* para definir a resposta da regra. Por fim, os consequentes de todas as regras passam por um processo de agregação, combinando os resultados individuais para formar um único conjunto *fuzzy* de saída.

- **Defuzzificação:** A saída da etapa de inferência é um conjunto *fuzzy*. No entanto, na maioria das aplicações, é necessário obter um valor numérico que será introduzido no processo (a corrente que passa por uma bobina ou a velocidade aplicada a um robô, por exemplo). Essa tradução de um conjunto *fuzzy* de saída em um valor preciso recebe o nome de Defuzzificação e é executada pelo método do centro de gravidade ou método da média dos máximos na grande parte das aplicações. No primeiro caso, a saída numérica é obtida por meio do ponto que divide a área abaixo da curva da função de pertinência do consequente em duas partes iguais. Já na segunda, o cálculo é realizado através da média dos dois pontos de maior valor da função de pertinência de saída.

Abordagens *fuzzy* são amplamente utilizadas, não somente no controle de sistemas robóticos, mas também em áreas como previsão do tempo [Setyanugraha et al. 2022], diagnóstico de doenças [Arji et al. 2019], mercado de ações financeiras [Nica, Delcea e Chiriță 2024] e reconhecimento de padrões faciais [Balovsyak et al. 2024]. Essa flexibilidade se dá devido à capacidade desta técnica em emular a tomada de decisões feita por humanos através de regras baseadas nas condicionais SE e ENTÃO. Outra vantagem desse tipo de técnica é a facilidade de entendimento do comportamento do processo, devido à interpretabilidade proporcionada pelas variáveis linguísticas. Um dos problemas inerentes a sistemas *fuzzy* é a construção do conjunto de regras, o que

muitas vezes requer um conhecimento prévio do funcionamento do processo como um todo. Caso contrário, a tomada de decisão pode não ser a ideal [Perfileeva 2007].

A capacidade dessa ferramenta de se adaptar a diferentes cenários levou à introdução de blocos *fuzzy* em conjunto com controladores clássicos, como o PID, que, isoladamente, não conseguem lidar com sistemas complexos e não lineares. Em uma aplicação onde um dos requerimentos do projeto é que o robô atinja uma determinada posição de maneira suave, sem grandes oscilações, sistemas baseados em lógica *fuzzy* são capazes de identificar o estado atual da planta e ponderar as ações de maneira adequada, ao passo que controladores PID com ganhos fixos, sintonizados por técnicas clássicas, não possuem esse tipo de maleabilidade, levando o robô a se mover em direção ao referencial muito rapidamente, o que pode causar um sobressinal elevado ou retardar a estabilização do mesmo [Carvajal, Chen e Ogmen 2000].

Abordagens híbridas *Fuzzy*-PID em robótica baseiam-se na modelagem da magnitude e da variação dos erros de estado por meio de funções de pertinência, que atribuem termos linguísticos às variáveis. O erro pode ser descrito, por exemplo, como ‘pequeno’, ‘médio’ ou ‘grande’, e sua variação como ‘positiva’, ‘negativa’ ou ‘nula’. A partir disso, um conjunto de regras é definido para cobrir todas as combinações possíveis das variáveis de entrada e ajustar dinamicamente os ganhos do controlador dentro de uma faixa determinada. Dessa forma, o robô adapta seu comportamento a diferentes condições operacionais, inclusive em situações anormais, como variações de erro causadas por distúrbios externos. A seguir, são apresentados trabalhos que utilizam lógica *fuzzy* na sintonização de controladores PID aplicados a drones.

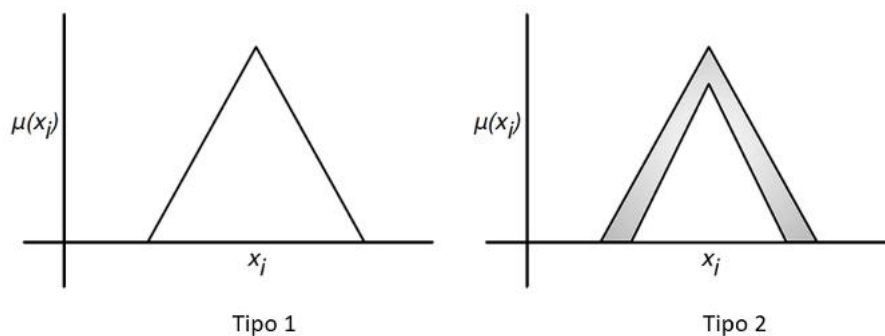
A primeira classe dos sistemas de lógica *fuzzy* é denominada tipo-1, sendo os controladores PID baseados nessa abordagem referidos neste trabalho como T1-PID. Rabah et al. [Rabah et al. 2019] aplicaram o T1-PID ao controle de movimento de um drone para rastreamento de alvos dinâmicos, demonstrando a eficácia da lógica *fuzzy* em trajetórias com comportamento imprevisível. Carvalho et al. [Carvalho et al. 2021] empregaram o T1-PID para estabilização de altitude, enquanto Melo et al. [Melo et al. 2022] ajustaram os ganhos do controlador com base na técnica clássica de Ziegler–Nichols. Por fim, Rao et al. [Rao et al. 2022]

integraram uma rede neural ao sistema *fuzzy* em uma metodologia de controle *offline* para drones.

Um problema enfrentado por sistemas baseados em lógica *fuzzy* do tipo-1 é a dificuldade em lidar com informações ruidosas, como valores faltantes ou leituras errôneas de sensores, devido à natureza fixa das funções de pertinência. Mesmo existindo uma variação no universo de atuação, elas não contemplam incertezas do modelo, uma vez que sistemas *fuzzy* do tipo-1 não são capazes de diferenciar ruído de dados úteis, o que pode afetar o desempenho dessas técnicas em aplicações nas quais nem sempre há precisão nos sistemas de aquisição de dados [Mittal et al. 2020].

Neste contexto, foram introduzidos os sistemas *fuzzy* do tipo 2, que se diferem da primeira geração através de conjuntos *fuzzy* do tipo 2 que contemplam uma região delimitada por várias funções fixas de tipo 1, as quais proporcionam à técnica uma maior maleabilidade em relação aos valores incertos dos antecedentes e consequentes [Mendel 2013]. A Figura 7 apresenta um exemplo de função de pertinência para sistemas de lógica *fuzzy* do tipo 1 e do tipo 2. A área sombreada no segundo caso é chamada de Pegada de Incerteza (do inglês, *Footprint of Uncertainty (FOU)*), a região na qual estão inseridas as incertezas das variáveis de entrada e de saída do sistema.

Figura 6 – Funções de pertinência de sistemas fuzzy do tipo 1 e do tipo 2.



Fonte: Adaptado de [Castanón-Puga et al. 2015].

A estrutura da lógica *fuzzy* do tipo 2 é similar a abordagem do tipo 1. A principal diferença está na substituição dos conjuntos *fuzzy* do tipo 1, que

são representados por funções singulares, por conjuntos do tipo 2, que englobam múltiplas funções sobrepostas. Além disso, é adicionada uma etapa de redução de tipo no final do processo para que a saída seja um valor exato, assim como em sistemas do tipo 1. A teoria da lógica *fuzzy* do tipo-2 foi apresentada em detalhes por Mendel et. al [Mendel e John 2002]. A seguir, serão apresentados os pontos principais necessários para a implementação desta técnica em sistemas digitais.

Um conjunto fuzzy do tipo 2 (\tilde{A}), caracterizado pela sua função de pertinência do tipo 2 $\mu_{\tilde{A}}(x, u)$, é representado da seguinte forma :

$$\tilde{A} = \left\{ \left((x, u), \underline{\mu}_{\tilde{A}}(x, u) \right) \mid \forall x \in X, \forall u \in J_x \subseteq [0, 1] \right\} \quad (2.1)$$

$$0 \leq \mu_{\tilde{A}}(x, u) \leq 1 \quad (2.2)$$

\tilde{A} pode ser reescrito como:

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} \frac{\mu_{\tilde{A}}(x, u)}{(x, u)}, J_x \subseteq [0, 1] \quad (2.3)$$

Em que $\int \int$ representa a união de todos os valores admissíveis para x e u . J_x é denotado como o conjunto primário de x , enquanto $\mu_{\tilde{A}}$ é um conjunto *fuzzy* do tipo 1, referido como o conjunto secundário que, na prática, é sempre unitário ($\mu_{\tilde{A}} = 1$).

A área compreendida pela união de todas as funções de pertinência primárias é conhecida como FOU. Em problemas práticos, a mesma deve ser projetada de forma que toda a incerteza dos estados de entrada e de saída seja contemplada nesta região, o que não era possível através de funções de pertinência singulares do tipo 1. A FOU pode ser definida como:

$$FOU(\tilde{A}) = \bigcup_{\forall x \in X} J_x = (x, u) : u \in J_x \subseteq [0, 1] \quad (2.4)$$

Outra representação usual para a FOU é apresentada por meio das funções de pertinência, que servem como limiares para a delimitação da região de incerteza. Essas funções recebem o nome de Funções de pertinência inferior $\underline{\mu}_{\tilde{A}}(x)$ (LMF) e superior $\underline{\mu}_{\tilde{A}}(x)$ (UMF), que podem ser definidas da seguinte forma:

$$\overline{\mu_{\tilde{A}}}(x) = \overline{FOU(\tilde{A})}, \forall x \in X \quad (2.5)$$

$$\underline{\mu_{\tilde{A}}}(x) = \underline{FOU(\tilde{A})}, \forall x \in X \quad (2.6)$$

O princípio de funcionamento da lógica *fuzzy* do tipo 2 é análogo ao do tipo 1, com algumas modificações. Na etapa de fuzzificação, os valores exatos de entrada são transformados em variáveis *fuzzy* enclausuradas pelas funções de pertinência LMF e UMF. A etapa de construção de regras não sofre alterações para sistemas do tipo 1. Para a inferência, como o conjunto *fuzzy* do tipo 2 é representado por um conglomerado de funções de pertinência, a força de ativação para i -ésima regra é dada por:

$$\tilde{F}_i = [\underline{f}^i, \overline{f}^i] \quad (2.7)$$

\underline{f}^i e \overline{f}^i são definidas pelas equações:

$$\underline{f}^i = \prod_{i=1}^n \underline{\mu_{\tilde{F}_n^i}} \quad (2.8)$$

$$\overline{f}^i = \prod_{i=1}^n \overline{\mu_{\tilde{F}_n^i}} \quad (2.9)$$

Um bloco extra é adicionado ao término da etapa de inferência para que as variáveis de saída, que inicialmente são do tipo 2, passem para valores exatos na saída. Existem diferentes técnicas para a redução de tipo em sistemas *fuzzy*. A grande parte delas baseiam-se no cálculo do centroide do conjunto de tipo 2 $C_{\tilde{A}}$ definido por:

$$C_{\tilde{A}} = \frac{1}{[y_l, y_r]} \quad (2.10)$$

y_l e y_r são calculados através das funções LMF e UMF, da seguinte forma:

$$y_l = \frac{\sum_{i=1}^L y_i \overline{\mu_{\tilde{A}}}(y_i | x') + \sum_{i=L+1}^N y_i \underline{\mu_{\tilde{A}}}(y_i | x')}{\sum_{i=1}^L \overline{\mu_{\tilde{A}}}(y_i | x') + \sum_{i=L+1}^N \underline{\mu_{\tilde{A}}}(y_i | x')} \quad (2.11)$$

$$y_r = \frac{\sum_{i=1}^R y_i \overline{\mu_{\tilde{A}}}(y_i | x') + \sum_{i=R+1}^N y_i \underline{\mu_{\tilde{A}}}(y_i | x')}{\sum_{i=1}^R \overline{\mu_{\tilde{A}}}(y_i | x') + \sum_{i=R+1}^N \underline{\mu_{\tilde{A}}}(y_i | x')} \quad (2.12)$$

Em que N é o número de pontos amostrados e L e R são pontos de troca de função. A metodologia para a determinação dos pontos delimitadores foi inicialmente proposta por Karnik et. al [Karnik e Mendel 2001], e vem recebendo melhorias ao longo do tempo visando a redução do esforço computacional [Wu e Mendel 2008]. Por fim, cada variável de saída passa por um processo de defuzzificação que define um valor exato através da média dos pontos y_l e y_r , ou seja:

$$y = \frac{y_l + y_r}{2} \quad (2.13)$$

As metodologias híbridas de lógica fuzzy do tipo 2 com PID são abreviadas, neste trabalho, pela sigla IT2-PID. O trabalho de [Sarabakha et al. 2017] empregou o IT2-PID para o controle de movimento de um drone, mesmo na presença de fortes rajadas de vento. Zhao et al. [Zhao et al. 2020] utilizou o IT2-PID para o controle de um robô que se acopla aos cabos de distribuição de energia para a tarefa de inspeção da rede. Já em [Candan et al. 2023] empregou o IT2-PID para o controle de um drone com capacidades de transportar cargas por meio de um cabo.

2.2.1.2 Controladores PID sintonizados por redes neurais

As redes neurais (RNs) são estruturas capazes de encontrar soluções para problemas por meio de otimização e aprendizado, processando informações através da comunicação entre neurônios organizados em subcamadas. Essas subcamadas fragmentam os dados em busca de padrões que maximizem uma função objetivo. Dessa forma, é natural considerar esse tipo de abordagem na busca por parâmetros para controladores, visando minimizar erros de estado e permitir que o sistema aprenda e se adapte a padrões previamente não observados.

A otimização dos ganhos de controladores PID por meio de redes neurais (RNs) ocorre por meio de uma etapa de treinamento, na qual a rede recebe dados de entrada do sistema e os processa utilizando pesos ajustáveis associados às conexões entre neurônios [Akhyar e Omatu 1993]. Durante o aprendizado, a rede extrai

informações sobre o comportamento dinâmico do sistema e sua resposta a diferentes sinais de entrada. Com isso, os parâmetros de ganho do controlador são ajustados iterativamente, de forma a minimizar a função de erro, que se aproxima de zero à medida que a rede converge para uma solução otimizada [Dev et al. 2020].

O trabalho de [Rosales, Soria e Rossomando 2019] empregou a técnica Neuro-PID para o controle de um drone hexarotor, no qual os estados do processo são estimados por meio do filtro de Kalman e inseridos na RN. Em [Chen et al. 2019], uma abordagem Neuro-PID foi utilizada em conjunto com o Algoritmo Genético, que tem por objetivo otimizar as funções de pesos da rede neural em uma aplicação de controle de posição de um drone. Uma abordagem similar foi proposta por [Wang et al. 2015] com a utilização de uma PSO em conjunto com a RN para a adaptação dos ganhos.

O presente trabalho utiliza um algoritmo de controle baseado na metodologia IT2-PID para a missão de inspeção das linhas de distribuição. A escolha se deu pela praticidade e eficiência de controladores PID em abordagens IBVS, além da facilidade de entendimento e robustez de sistemas de lógica *fuzzy*. A utilização da lógica *fuzzy* do tipo 2 foi baseada na necessidade de se levar em conta as incertezas atreladas ao processo de posicionamento por meio da identificação de objetos em imagens e nas medições do sistema sensorial do drone.

2.3 Trabalhos Correlatos

Larrauri et al. [Larrauri, Sorrosal e González 2013] propuseram um sistema de inspeção de linhas capaz de detectar a distância de árvores, vegetação e construções próximas às linhas de distribuição e enviar relatórios por mensagens de e-mail e SMS. A trajetória a ser seguida é feita de forma off-line (previamente a missão) por meio de dados de GPS, podendo ser editada durante o voo. A detecção dos cabos de energia é realizada por meio da transformada de Hough em imagens de uma câmera óptica e o controle do drone é realizado por um Modelo de controle preditivo (MCP).

Araar et al. [Araar e Aouf 2014] apresentaram uma metodologia de seguimento de linhas de distribuição baseada em um controle posicional relativo ao

plano bi-dimensional da imagem. As linhas foram detectadas pela transformada de Hough, e dois controladores PID foram empregados para o alinhamento lateral e angular. Apesar de os resultados terem sido obtidos em um ambiente controlado, sem levar em conta fatores como iluminação comprometida e o efeito do vento, este é um dos primeiros trabalhos a considerar o problema de seguimento de linhas por meio de técnicas visuais.

Valipour et al. [Valipour, Khandani e Moradi 2015] projetaram um Controlador *backstepping* para a tarefa de seguimento das linhas de energia em um ambiente simulado. A detecção foi feita através da transformada de Hough e a estimação da posição do drone foi realizada por meio de um filtro de Kalman.

Deng et al. [Deng et al. 2016] propuseram um sistema de inspeção de linhas de transmissão em um ambiente real, no qual a posição das linhas é obtida de maneira global por meio de dados de GPS e localmente através de medições de um sensor LiDAR processadas por um filtro de Kalman. O Controle do drone é realizado somente no âmbito global por meio de *waypoints* das torres obtidos por GPS.

Jeong et al. [Jeong et al. 2018] abordaram um sistema para inspeção de linhas de distribuição no qual a transformada de Hough é empregada junto com a técnica de agrupamento k-means para a remoção de elementos retilíneos incorretos e, em seguida, empregaram um controlador PD para o alinhamento do drone, apresentando resultados em um ambiente controlado.

Solilo et al. [Solilo e Doorsamy 2018] propuseram uma metodologia para a inspeção de linhas de distribuição com desvio de obstáculos, com resultados obtidos em ambiente controlado. Os cabos são detectados através da aplicação da segmentação por *thresholding* no espaço de cores HSV, que possibilita a obtenção de uma imagem binária contendo os *pixels* correspondentes as linhas. A detecção dos obstáculos é feita por meio de um sensor de distância ultrassônico.

Miralles et al. [Mirallès et al. 2018] propuseram uma metodologia de inspeção semi-assistida, na qual um operador é responsável por controlar manualmente a altitude e a velocidade de avanço do drone. O alinhamento lateral e angular com os cabos de energia é realizado automaticamente por meio de dois controladores PBVS desacoplados, que utilizam a posição estimada do drone fornecida por um filtro de

Kalman. Esse filtro combina leituras de um sensor LiDAR com um filtro rígido aplicado à imagem para detectar as bordas das linhas. Além disso, o operador pode ativar a função de pouso nos cabos, que é facilitada pelo algoritmo de alinhamento.

Ceron et al. [Cerón, Mondragón e Prieto 2018] propuseram um método de detecção de linhas que recebeu o nome de CBS (do inglês, *Circle based search*). Comparações com outras técnicas baseadas em gradientes, como a Transformada de Hough e o LSD, mostraram a superioridade da técnica. Além disso, foi proposta uma missão de inspeção de linhas de distribuição em cenários reais, na qual controladores proporcionais são utilizados para o controle lateral e angular, sendo que os valores de orientação dos cabos passam por um filtro Média-móvel para a remoção de medidas ruidosas.

Menendez et al. [Menéndez, Pérez e Cheein 2019] propuseram um sistema para a detecção e inspeção das linhas, no qual informações como a distância em *pixels* entre os cabos e a altura relativa do drone às linhas são obtidas em um estágio prévio à missão de seguimento das linhas. Essas informações são utilizadas posteriormente, em conjunto com as imagens pós-processadas, para a identificação das linhas através da transformada de Hough. As informações visuais são utilizadas em conjunto com dados de sensores IMU e GPS para a estimativa precisa da posição do drone. A metodologia foi testada em um cenário real.

Rafique et al. [Rafique, Fu e Mai 2021] propuseram uma metodologia generalizada para o seguimento de estruturas retilíneas, dentre as quais as linhas de distribuição estão inseridas. Para o controle do drone, é utilizado um controlador PID, enquanto as linhas são detectadas através da aplicação de limiares no espaço de cores RGB.

Takaya et al. [Takaya et al. 2021] apresentaram um sistema de navegação de drones capaz de inspecionar linhas de distribuição, no qual o ponto inicial da trajetória é escolhido por meio de um GPS (controlador global), e um controlador PID é utilizado para o controle local, em que a posição das linhas é obtida por meio de dados de um sensor LiDAR. Uma câmera acoplada ao drone auxilia o operador a detectar possíveis falhas no sistema de forma manual. Os resultados foram obtidos em cenários reais.

Diniz et al. [Diniz et al. 2022] propuseram um sistema para auxílio aos ope-

radores em uma tarefa de inspeção das linhas de energia. Os cabos são identificados através da rede YOLOv4, e a *bounding box* do objeto detectado serve como uma região delimitadora para a aplicação da Transformada de Hough, a fim de detectar a localização e a orientação das linhas. Além disso, foi elaborada uma interface gráfica na qual o drone pode ser posicionado próximo à posição inicial da missão, realizar o seguimento das linhas de energia através de um controlador proporcional e realizar o pouso nas linhas de maneira semi-autônoma.

Li et al. [Li et al. 2023] apresentaram uma metodologia para inspeção de linhas de alta tensão que inclui dois tipos de operações: a primeira prevê a aquisição de imagens com o drone acima das linhas, enquanto a segunda promove a inspeção paralela, na qual o drone segue os cabos em diferentes níveis de altitude. Ambos os testes foram realizados em cenários reais. As linhas são identificadas por meio da rede YOLOX, e o controle local é realizado por meio de um Controlador de modo deslizante (do inglês, *Sliding mode controller* (SMC)). Além disso, é realizada a detecção de ninhos de pássaros nos postes conforme o drone transita pelas linhas.

As informações dos trabalhos apresentados nesta seção são resumidas pela Tabela 2. A metodologia proposta utiliza o YOLOv8 para a tarefa de segmentação da área compreendida pelas linhas de distribuição, além da detecção dos postes de energia, em conjunto com a cruzeta e isoladores. O treinamento da rede foi realizado com imagens contendo diferentes tipos de configurações de linhas, com orientações e iluminações variadas, permitindo que a inspeção seja realizada em diferentes cenários. O controle foi escolhido levando em conta a capacidade dos modelos *fuzzy* de se adaptar a diferentes estados da planta, além de lidar com ruídos na estimativa dos parâmetros.

2.3.1 Conclusões Parciais

Este capítulo apresentou o referencial teórico dos pontos principais necessários para o entendimento da metodologia proposta neste trabalho. A seção 2.1 tratou das técnicas empregadas para a identificação dos cabos de energia e de outros elementos do sistema elétrico, mostrando suas vantagens e desvantagens, além da empregabilidade das mesmas em trabalhos envolvendo drones. Em seguida, a Seção 2.2 abordou diferentes métodos de sintonização de ganhos em controladores

Tabela 2 – Trabalhos correlatos

Autor	Método detec.	Controlador	Detecção de Elementos	Resultados
Larrauri et al. [Larrauri, Sorrosal e González 2013]	Hough	MCP	Não	Ambiente real
Araar et al. [Araar e Aouf 2014]	Hough	PID	Não	Ambiente controlado
Valipour et al. [Valipour, Khandani e Moradi 2015]	Hough	Controlador <i>backstepping</i>	Não	Simulação / Amb. controlado
Deng et al.[Deng et al. 2016]	LiDAR	Controle por <i>waypoints</i>	Não	Ambiente real
Jeong et al. [Jeong et al. 2018]	Hough	PD	Não	Ambiente controlado
Solilo et al. [Solilo e Doorsamy 2018]	Segmentação por <i>thresholding</i>	PID	Não	Ambiente controlado
Mirales et al. [Mirallès et al. 2018]	LiDAR + Filtro rígido	PI	Não	Ambiente real
Ceron et al. [Cerón, Mondragón e Prieto 2018]	CBS	P	Não	Ambiente real
Menendez et al. [Menéndez, Pérez e Cheein 2019]	Hough	Controle por <i>waypoints</i>	Não	Ambiente real
Rafique et al. [Rafique, Fu e Mai 2021]	Segmentação por <i>thresholding</i>	PID	Não	Ambiente controlado
Takaya et al. [Takaya et al. 2021]	LiDAR	PID	Não	Ambiente real
Diniz et al. [Diniz et al. 2022]	YOLOv4	P	Não	Simulação / Amb. controlado
Li et al. [Li et al. 2023]	YOLOx	SMC	Sim	Ambiente real
Presente Trabalho	YOLOv8-Seg	IT2-PID	Sim	Simulação / Amb. Real

PID, bem como a aplicação das técnicas apresentadas para o controle posicional e de velocidade de um drone. Por fim, foi apresentado um compilado de trabalhos relacionados a metodologias para inspeção de linhas de energia, abordando as técnicas utilizadas para a identificação das linhas e o método de controle empregado, com a inserção das técnicas empregadas neste trabalho.

Este trabalho insere-se no contexto do planejamento de trajetórias locais, em que não há informações prévias sobre o trajeto a ser inspecionado. Os dados da missão podem ser acessados em tempo real ou armazenados na máquina responsável pela comunicação com o drone, possibilitando uma tomada de decisão mais precisa durante a operação. Além disso, o detector de objetos utilizado pode ser expandido para identificar falhas comuns em linhas de transmissão. A integração de sensores adicionais também pode aumentar a robustez do sistema, mitigando possíveis falhas na visão computacional que poderiam comprometer o desempenho do controle do drone.

3 Metodologia Proposta

Este capítulo tem como objetivo apresentar o detalhamento dos processos necessários para a implementação da metodologia de inspeção de linhas de distribuição. Esta pesquisa tem como foco propor uma abordagem de planejamento de trajetória local, na qual não há conhecimento prévio dos pontos a serem percorridos pelo drone. Dessa forma, as missões de inspeção consideradas neste trabalho são estruturadas em quatro etapas: escolha do tipo de missão, posicionamento manual para o início da operação, execução da tarefa e comando de término da missão.

Inicialmente, o operador deve selecionar a tarefa a ser executada pelo drone. As três opções disponíveis são: alinhamento lateral, alinhamento angular e seguimento, conforme ilustrado na Figura 7. As velocidades v_y e ω_y são determinadas pelos controladores IT2_PID e estão associadas, respectivamente, ao alinhamento lateral e ao alinhamento angular do drone em relação às linhas. A velocidade v_x corresponde ao movimento de avanço ao longo da trajetória e é definida de acordo com os requisitos da missão de inspeção.

A operação do VANT é iniciada com o controle manual pelo operador, que conduz o drone a partir de um ponto em solo até uma posição próxima às linhas de distribuição, definida como o ponto inicial da missão. A partir desse ponto, o modo automático é ativado, e o drone passa a utilizar exclusivamente as informações visuais para guiar seu movimento, sem a necessidade de um planejamento global de trajetória. A altura de operação é mantida aproximadamente constante por meio do controle interno do veículo, sendo definida previamente de acordo com as condições da missão.

Durante a execução da tarefa de seguimento, o drone avança ao longo das linhas com velocidade linear constante de aproximadamente 1m/s e mantendo a altitude de 13.5m em relação ao solo, enquanto os controladores atuam para corrigir desvios laterais e angulares em relação ao objeto de interesse. A definição dessa altitude leva em consideração não apenas aspectos de segurança operacional e qualidade de visualização das linhas, mas também a mitigação de efeitos de interferência eletromagnética provenientes da infraestrutura de transmissão, os quais podem impactar a qualidade da comunicação, incluindo o envio do fluxo de

imagens e dos sinais de controle entre o drone e a estação em solo.

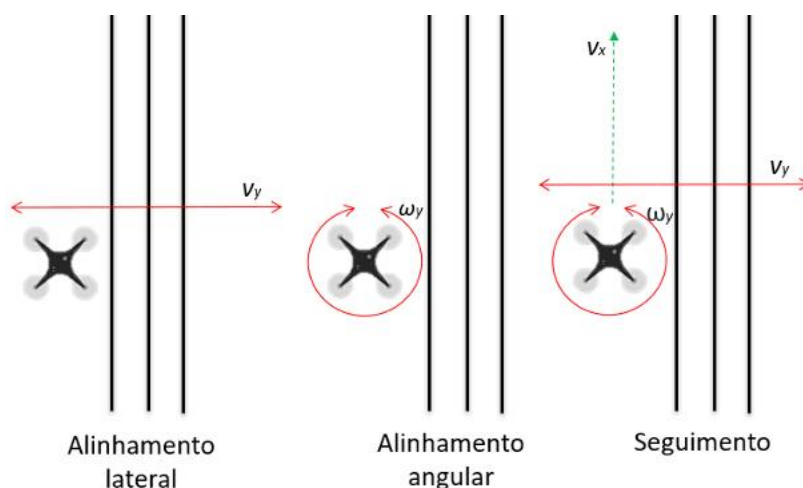
A trajetória não é previamente definida, sendo estimada em tempo quase real a partir das informações extraídas das imagens, caracterizando uma estratégia de navegação reativa baseada em visão.

O término da missão é definido pelo operador, que pode, a qualquer momento, interromper o modo automático e retomar o controle manual do drone. Essa abordagem garante maior flexibilidade operacional e segurança durante a execução das inspeções.

- Alinhamento lateral: Nesta tarefa, o drone deve ser posicionado a uma distância determinada da porção central delimitada pelos cabos das extremidades. Usualmente, em sistemas de controle IBVS, adota-se a região central da imagem como referencial e, portanto, se a câmera estiver posicionada no centro longitudinal do corpo do drone, isso indica que ele deve manter-se acima dos cabos.
- Alinhamento angular: Nesta tarefa, o drone deve posicionar-se de forma que a orientação relativa dos cabos se mantenha de acordo com um referencial determinado. Geralmente, adota-se um referencial nulo e, como o vetor que passa pelo centro da lente é coincidente com o centro geométrico do robô, esse referencial indica que o drone deve posicionar-se paralelamente aos cabos.
- Seguimento: Para essa tarefa, é aplicada uma velocidade de avanço linear, e o drone deve se manter alinhado tanto em relação à posição lateral quanto à orientação angular enquanto se movimenta ao longo das linhas. Caso exista a interseção entre duas ou mais torres que leve a uma mudança na direção dos cabos, o drone deve ser capaz de detectar a variação no perfil da trajetória e ajustar os comandos de controle de forma que o referencial não seja perdido. Durante o seguimento, o sistema é capaz de detectar automaticamente elementos da rede de distribuição presentes no campo de visão da câmera, fornecendo ao operador informações visuais relevantes para a inspeção. No entanto, a identificação de falhas ou anomalias nesses componentes não é realizada de forma automática, ficando sob responsabilidade do operador a análise e a tomada de decisão quanto à necessidade de uma inspeção mais detalhada.

Além disso, neste modo, o operador tem a liberdade de pausar o seguimento para realizar uma inspeção manual de um ponto específico e, em seguida, optar pela continuidade ou o término da missão.

Figura 7 – Tipos de missão abordados neste trabalho



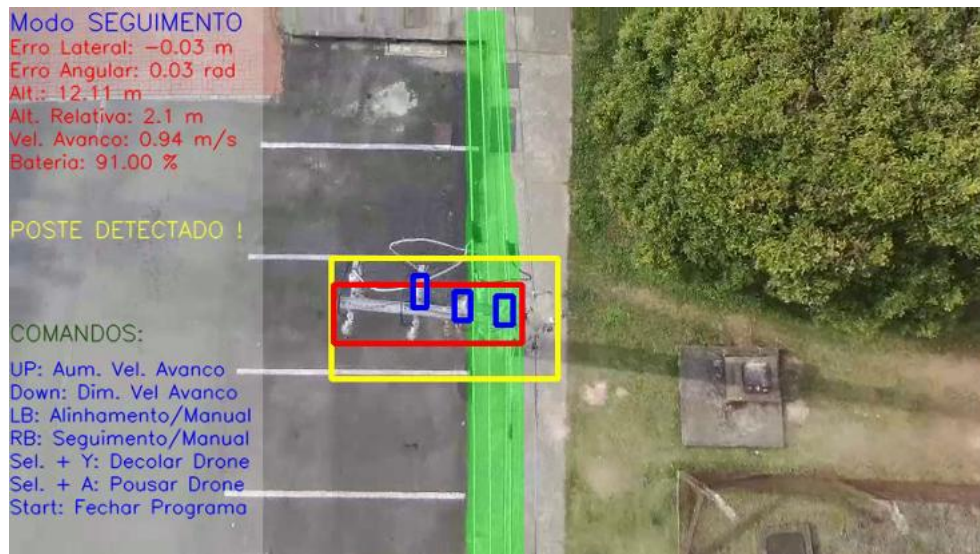
Fonte: Elaborado pelo autor.

Após a escolha do tipo de missão a ser executada, o operador deve decolar o drone até que ele atinja um ponto acima dos cabos de distribuição. A operação é realizada com o auxílio de um computador portátil que se comunica com o drone, recebendo informações de telemetria e imagens da câmera acoplada em sua estrutura, e envia sinais para o controle do mesmo. É importante salientar que esta comunicação é de suma importância para o andamento e segurança das missões, bem como para os equipamentos e os operadores. Qualquer falha na transmissão das imagens, por exemplo, pode levar ao cálculo errôneo da posição do drone, afetando a movimentação do mesmo. Desta forma, as informações, como nível de bateria, altitude, fluxo de imagens e velocidades do drone, devem estar disponíveis para o operador, que tem a capacidade de retornar rapidamente ao modo de controle manual em caso de alguma anomalia.

Para o início da missão, deve se escolher uma posição na qual as linhas estejam no campo de visão da câmera, com uma altura adequada para evitar colisões

e interferências. Caso contrário, o sistema de controle baseado em informações visuais não consegue determinar a posição do objeto a ser seguido, e a malha de controle se torna incapaz de calcular as ações a serem tomadas. Sendo respeitadas essas condições, o operador deve acionar um comando para o início da missão. Um ponto de interesse na inspeção da rede elétrica de energia é os postes de distribuição de energia. Pensando nisso, foi introduzido um modelo capaz de detectar esses elementos em conjunto com objetos menores, como as cruzetas e isoladores elétricos. Desta forma, além das informações do drone e das linhas de distribuição, o operador é alertado visualmente caso esteja passando pelos postes e, em uma missão de seguimento, tem a possibilidade de manter o seguimento autônomo ou passar para o modo de inspeção manual do poste e dos outros elementos, tendo a possibilidade de continuar o seguimento ou finalizar a missão. Um exemplo da tela mostrada ao operador durante o voo do drone pode ser visto na Figura 8. No canto esquerdo estão as informações de comando e telemetria do drone, bem como a distância do drone para às linhas, além de um alerta caso algum poste seja detectado na imagem. A região em verde corresponde à segmentação das linhas, e os retângulos amarelos, vermelhos e azuis indicam a detecção dos postes, cruzetas e isoladores, respectivamente.

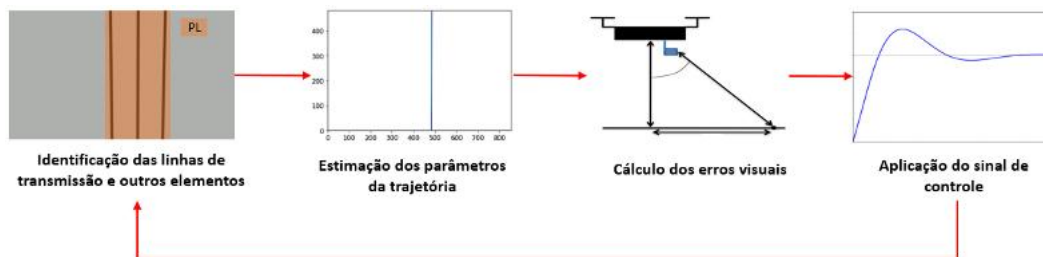
Figura 8 – Tela da missão



Fonte: Elaborado pelo autor.

Inicialmente, a metodologia deve ser capaz de identificar a região delimitada pelos cabos na imagem; esse processo é realizado pela rede YOLOv8 com a tarefa de segmentação e os outros elementos por meio do YOLOv8 através de um modelo de detecção de objetos. Em seguida, é realizada a parametrização da área compreendida entre os cabos da extremidade por meio de uma técnica de ajuste de curvas. Com as informações sobre a região do objeto mapeadas em termos de coordenadas no plano da imagem, o próximo passo é determinar os erros visuais relacionados à posição atual do objeto e o referencial adotado. Por fim, esses valores de erro são transmitidos para o bloco *fuzzy* que calcula os ganhos utilizados pelos controladores IT2-PID. Esses processos são executados pela estação em solo até que o operador envie um comando ao drone para o fim da missão. A Figura 9 resume os processos da metodologia proposta para a inspeção das linhas. As Seções 3.1, 3.3, 3.4 e 3.5 tratarão de cada etapa de forma detalhada.

Figura 9 – Esquemático dos passos referentes a metodologia proposta



Fonte: Elaborado pelo autor.

3.1 Detecção das linhas de distribuição

Abordagens IBVS requerem que o objeto a ser seguido seja bem definido durante o teste. Caso contrário, o sistema de controle pode perder o referencial, comprometendo a integridade da missão, bem como a dos operadores e dos robôs. As linhas de distribuição em imagens aéreas geralmente apresentam uma forma retilínea que pode ser confundida com outros elementos em cenários urbanos, como construções de concreto e marcações no asfalto. Além disso, o fundo abaixo das linhas apresenta padrões e cores variadas, e o nível de iluminação do ambiente afeta a saturação dos cabos. Assim, é de suma importância que a técnica utilizada para detecção tenha a capacidade de distinguir os elementos de interesse sem que objetos espúrios prejudiquem os resultados.

A detecção de objetos por redes neurais baseia-se, inicialmente, em uma etapa de aprendizado, na qual imagens contendo anotações sobre quais objetos existem nas imagens e a localização dos mesmos são utilizadas para treinar o modelo, permitindo que ele aprenda a reconhecer e localizar esses objetos em novas imagens. Conforme mais imagens transitam pelas camadas da rede, os pesos das conexões são atualizados, o que, na prática, significa que o modelo passa a ter a capacidade de distinguir certos padrões de uma determinada classe de objetos. Essa etapa é chamada de treinamento da rede. O desempenho de um modelo detector de objetos está relacionado, sobretudo, com a qualidade e variabilidade das imagens que são utilizadas para o treinamento da rede [Dhillon e Verma 2020]. Desta forma, uma

prática usual é a escolha de imagens contendo os objetos em diferentes tamanhos, orientações e níveis de iluminação do ambiente. Para a detecção de carros, por exemplo, vários modelos devem ser utilizados, de forma que a rede adquira a capacidade de generalizar a identificação, mesmo para modelos que não foram inseridos na etapa de treinamento.

Ao término da etapa de treinamento da rede, é gerado um arquivo contendo os pesos otimizados do modelo, que indicam os parâmetros ajustados durante a etapa de aprendizado. Este arquivo pode ser utilizado para a realização de inferências futuras, sem que todo o processo de aprendizado, que costuma ser demorado, precise ser executado novamente. Por padrão, o YOLOv8 é pré-treinado por meio do banco de imagens COCO *dataset* [Lin et al. 2014], um *dataset* que contém 2.5 milhões de objetos anotados de 91 classes distintas em 328.000 imagens. Ao analisar a composição do banco de imagens, é possível observar que o foco deste está na detecção de objetos cotidianos, como pessoas, veículos, animais de estimação e utensílios domésticos. A classe de linhas de distribuição não está presente e, assim, a rede não é capaz de identificá-las com os pesos padrões disponibilizados na instalação da rede.

Uma saída para este problema é a realização do treinamento do modelo de detecção através de um banco de imagens personalizado, contendo apenas a classe de linhas de energia. A seguir, são apresentadas as etapas de aquisição de dados, anotação de imagens e treinamento do YOLOv8, realizadas para a obtenção do modelo de detecção e segmentação de linhas de distribuição utilizado neste trabalho.

Inicialmente, foi construído um banco de imagens contendo linhas de distribuição de energia. Para tal, foram escolhidas imagens autorais, contendo diferentes tipos de configurações de cabos, orientações, tipos de fundo e condições de iluminação obtidas através de sobrevoos com um drone em diversas localidades. Além disso, para aumentar a generalização do modelo, foram escolhidas algumas imagens retiradas de bancos de imagens dos trabalhos de [Zhang et al. 2019] e [Abdelfattah, Wang e Wang 2020], com foco em cenários nos quais a câmera está completamente acima das linhas, de forma a contemplar as situações que o drone encontrará nas missões de inspeção abordadas neste trabalho. Desta forma, o

banco de imagens construído possui 1385 imagens aéreas, 1265 das quais contêm linhas de distribuição e transmissão de energia, e 123 contendo apenas ambientes sobrevoados sem nenhuma linha observada, que foram introduzidas com o objetivo de reduzir a detecção de falsos positivos.

O próximo passo necessário para a construção do banco de imagens personalizado é a anotação das imagens. Cada uma delas passa por um observador que determina a existência do objeto na imagem e em que região ele se encontra. Existem diversos *softwares* de código aberto para esta tarefa, e neste trabalho foi escolhido o Roboflow [Dwyer et al. 2022], uma plataforma que dispõe de várias ferramentas para anotação de imagens em diversos formatos, hospedagem do banco de imagens, criação de imagens sintéticas e o treinamento da rede. Como a rede é utilizada para a segmentação do objeto na imagem, a região compreendida pelos cabos deve ser anotada com precisão. Caso contrário, a segmentação pode ser afetada, mesmo que o objeto seja detectado. Pensando nisso, foi escolhida uma ferramenta de anotação poligonal, na qual a área do objeto pode ser delimitada por infinitos pontos, em detrimento das técnicas de anotação retangulares, que podem considerar regiões fora do escopo do objeto, prejudicando o processo de aprendizado da rede. A Figura 10 mostra a aplicação dessas duas técnicas de anotação, exemplificando o problema descrito.

Figura 10 – Exemplo de anotações de imagens em cenários contendo linhas de distribuição



Fonte: Elaborado pelo autor.

Ao término da etapa de anotação, existe a possibilidade de criação de imagens sintéticas. Este processo é importante para que o modelo seja capaz de identificar objetos em cenários não contemplados no banco de imagens original. Pensando nisso, duas questões foram formuladas, para a geração de cenários artificiais: o modelo deve ser capaz de detectar as linhas independentemente da orientação do drone e, além disso, fatores como a estação do ano e a hora do dia que afetam o nível de iluminação solar não devem exercer grande influência nos resultados. Assim, para cada imagem original, foram criadas duas imagens sintéticas com as seguintes transformações aplicadas de maneira aleatória:

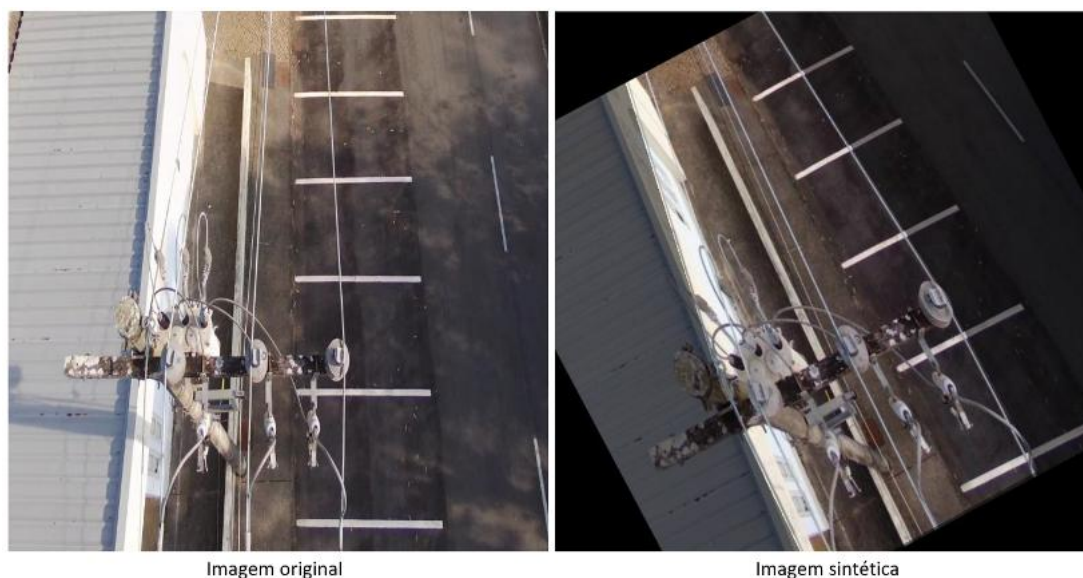
- Rotação: Entre -45° e 45°
- Brilho: Entre -25% e $+25\%$

A Figura 11 apresenta um exemplo de imagem sintética gerada a partir de uma das imagens originais do conjunto. Ao final deste processo, o banco de

imagens final utilizado para o treinamento da YOLOv8-seg passa a conter 3318 imagens, sendo dessas:

- Conjunto de treinamento: 2901 imagens
- Conjunto de validação: 263 imagens
- Conjunto de teste: 154 imagens

Figura 11 – Exemplo de imagem sintética obtida através de uma imagem do *dataset* original.

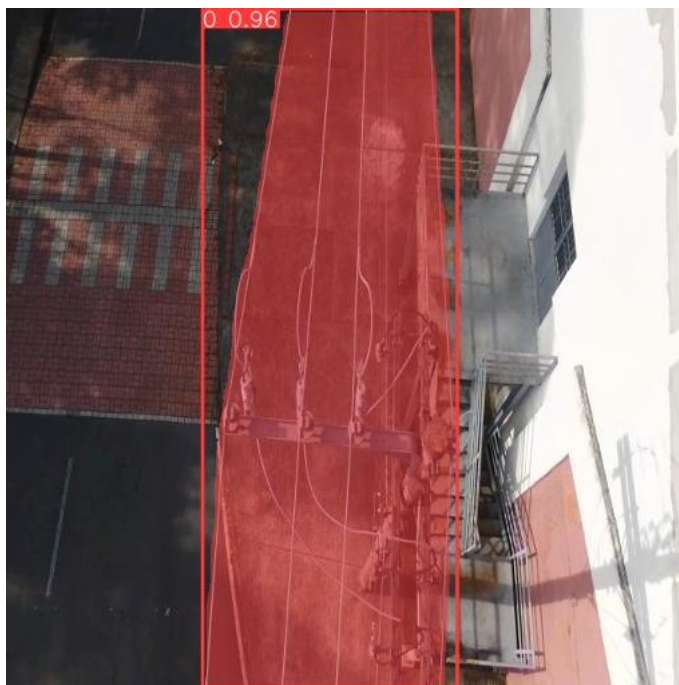


Fonte: Elaborado pelo autor.

Com as imagens devidamente anotadas, o próximo passo é a realização do treinamento do modelo, o qual foi feito por meio do ambiente de programação Google Colab, em linguagem Python, podendo ser acessado pelo *link*. É importante salientar que essa etapa é feita de maneira *offline*, antes das missões de inspeção, uma vez que é um processo demorado que depende de fatores como o tamanho e a quantidade de imagens, o modelo da rede e a capacidade computacional. Assim, no final do processo, é gerado um arquivo no formato '.pt' que possui os pesos de cada

camada da rede e é utilizado para as inferências posteriores, através das imagens de entrada provenientes da câmera do drone. A imagem 12 apresenta um exemplo de inferência por meio dos pesos obtidos na etapa de treinamento. O retângulo indica o objeto detectado pelo modelo. Acima deste, são apresentados dois termos: o primeiro refere-se à identificação da classe do objeto (neste exemplo, '0' refere-se à classe das linhas de distribuição). Já o segundo termo (neste exemplo, 0.96) indica a probabilidade estimada de que o objeto pertença à classe prevista. A saída do YOLOv8-seg exibe tanto a *bounding box* quanto a *núvem de pixels* correspondente ao objeto.

Figura 12 – Exemplo de inferência através da YOLOv8-seg



Fonte: Elaborado pelo autor.

3.2 Detecção dos elementos do sistema de distribuição de energia

Além dos cabos que conduzem a energia, outros itens são cruciais para garantir a qualidade da energia a ser distribuída. Assim, os drones utilizados para as inspeções da rede elétrica devem ser capazes de identificar esses elementos de

forma a observar eventuais problemas por meio de inspeções visuais, como corrosão, acúmulo de sujeira, presença de ninhos de pássaros ou obstrução por galhos de árvores. Devido à qualidade da transmissão da imagem do drone Bebop 2 à central que realiza o processamento das imagens ser reduzida, o foco deste trabalho é a detecção dos elementos mais comuns na malha de energia, como os postes, as cruzetas e os isoladores, sem a detecção de possíveis falhas nesses itens, que pode ser feita pelo operador ao término da missão, com o auxílio de imagens com maior resolução que são armazenadas internamente no drone.

Para a tarefa de detecção desses elementos, utiliza-se o YOLOv8 com um modelo de detecção de objetos. Assim como para a segmentação das linhas de distribuição, é necessária a construção de um banco de imagens com os itens anotados para o treinamento da rede. Assim, foram adquiridas 564 imagens; dentre essas, 494 contém postes de distribuição em conjunto com os outros elementos menores, e 70 imagens sem nenhum objeto. Para aumentar a variabilidade, também foram introduzidas imagens artificiais no conjunto de treinamento, com as mesmas transformações aplicadas ao conjunto de dados das linhas de distribuição, de forma que o dataset final tenha 1364 imagens nos seguintes conjuntos :

- Conjunto de treinamento: 1200 imagens
- Conjunto de validação: 110 imagens
- Conjunto de teste: 54 imagens

Um exemplo de inferência pode ser visto na Figura 13. O retângulo verde indica o poste, o amarelo indica a cruzeta, enquanto os vermelhos mostram os isoladores.

Figura 13 – Exemplo de inferência dos elementos do sistema elétrico com YOLOv8



Fonte: Elaborado pelo autor.

3.3 Estimativa dos parâmetros da trajetória

A partir da máscara de segmentação obtida pela saída da YOLOv8-seg, é possível computar uma trajetória que será utilizada como referência para o drone executar as missões de inspeção. A estratégia utilizada neste trabalho funciona da seguinte forma: inicialmente, é criada uma imagem com os *pixels* assumindo valores nulos nos três canais do espaço de cores RGB (0, 0, 0). Em seguida, o conjunto de *pixels* compreendido pela máscara de pontos é adicionado a essa imagem nula com o valor máximo dos canais (255, 255, 255), em um processo análogo ao de colar figuras em uma folha de papel. O resultado da operação é chamado de máscara binária, na qual os *pixels* em branco representam a região compreendida pelo objeto, conforme mostrado na Figura 14. Com isso, o próximo passo é traduzir os pontos do objeto em uma curva parametrizada que permita a obtenção dos parâmetros necessários para o cálculo dos erros visuais que nutrem a malha de controle responsável por manter o drone na posição determinada.

Em imagens aéreas, a área ocupada pelos cabos de energia geralmente pode ser representada por um quadrilátero. No entanto, em certos cenários, a presença de um poste pode alterar a direção dos cabos, modificando sua orientação original.

Desta forma, a obtenção dos erros visuais neste trabalho é realizada por meio da parametrização da região compreendida pelas linhas de distribuição, através de uma regressão quadrática. A Figura 14 apresenta um exemplo de um cenário que justifica o uso de equações de ordens superiores para a estimativa dos parâmetros da trajetória. Neste exemplo, uma parametrização linear não seria capaz de representar a mudança de orientação dos cabos, podendo gerar imprecisões no cálculo dos erros visuais e, conseqüentemente, afetar o desempenho da malha de controle.

Figura 14 – Exemplo de máscara binária em cenário com mudança de orientação das linhas.

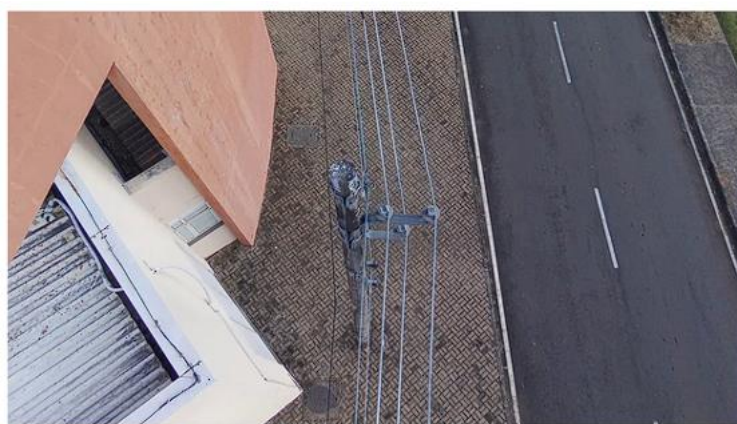


Imagem original



Máscara binária

Fonte: Elaborado pelo autor.

O método escolhido para a estimação dos parâmetros do modelo é o Mínimos

Quadrados Ordinários (do inglês, Ordinary Least Squares ou OLS). Inicialmente, é feita uma varredura da máscara binária em busca dos *pixels* não nulos, e as coordenadas (altura, comprimento) desses pontos são armazenadas em um vetor de dados B , que passa pelo regressor OLS, cujo princípio de funcionamento é apresentado a seguir:

Dada uma equação quadrática do tipo:

$$x = a_0 + a_1y + a_2y^2 \quad (3.1)$$

Onde x é o ponto longitudinal da trajetória que, na abordagem IBVS, corresponde ao valor do comprimento da imagem, em *pixels*, e y é a coordenada latitudinal que indica o valor da altura da imagem. Um problema de mínimos quadrados pode ser entendido como a minimização da soma dos quadrados da equação residual $r(y)$:

$$\min r(y) = \min \sum_{i=0}^{n-1} (a_0 + a_1y_i + a_2y_i^2 - x_i)^2 \quad (3.2)$$

Em que n é o número de elementos do vetor B . Para que $A = [a_0, a_1, a_2]$ seja um vetor candidato à solução do problema de minimização, a seguinte condição deve ser respeitada [Hansen, Pereyra e Scherer 2013] :

$$\nabla \rho(y) = \frac{\partial \rho(y)}{\partial a_k} = 0, k = 0, 1, 2 \quad (3.3)$$

Dessa forma, as equações diferenciais parciais para cada coeficiente podem ser obtidas por meio das equações :

$$\frac{\partial \rho}{\partial a_0} = 2 \sum_{i=0}^{n-1} (a_0 + a_1y_i + a_2y_i^2 - x_i) = 0 \quad (3.4)$$

$$\frac{\partial \rho}{\partial a_1} = 2 \sum_{i=0}^{n-1} (a_0y_i + a_1y_i^2 + a_2y_i^3 - x_iy_i) = 0 \quad (3.5)$$

$$\frac{\partial \rho}{\partial a_2} = 2 \sum_{i=0}^{n-1} (a_0y_i^2 + a_1y_i^3 + a_2y_i^4 - x_iy_i^2) = 0 \quad (3.6)$$

Reorganizando as equações através da eliminação dos somatórios e da remoção dos termos constantes, quando possível, tem-se que:

$$a_0 n + a_1 \sum_{i=0}^n y_i + a_2 \sum_{i=0}^n y_i^2 = \sum_{i=0}^n x_i \quad (3.7)$$

$$a_0 \sum_{i=0}^n y_i + a_1 \sum_{i=0}^n y_i^2 + a_2 \sum_{i=0}^n y_i^3 = \sum_{i=0}^n y_i x_i \quad (3.8)$$

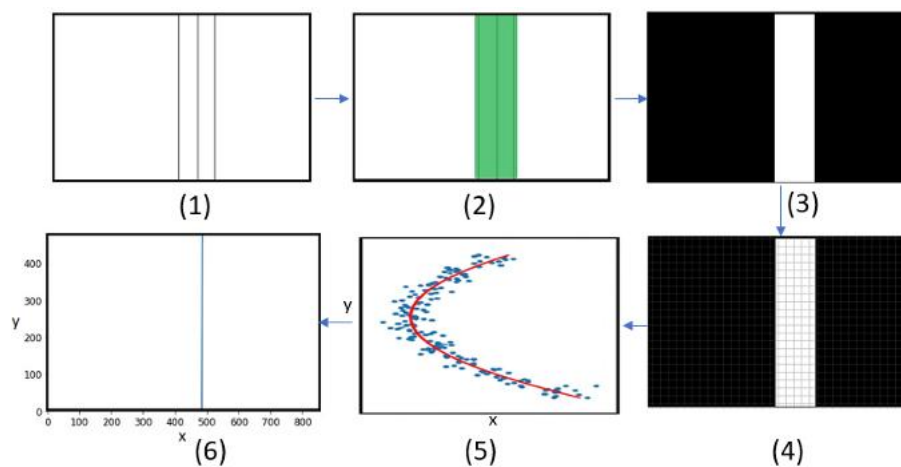
$$a_0 \sum_{i=0}^n y_i^2 + a_1 \sum_{i=0}^n y_i^3 + a_2 \sum_{i=0}^n y_i^4 = \sum_{i=0}^n y_i^2 x_i \quad (3.9)$$

As equações acima podem ser reescritas na forma matricial da seguinte maneira:

$$\begin{bmatrix} n & \sum_{i=0}^n y_i & \sum_{i=0}^n y_i^2 \\ \sum_{i=0}^n y_i & \sum_{i=0}^n y_i^2 & \sum_{i=0}^n y_i^3 \\ \sum_{i=0}^n y_i^2 & \sum_{i=0}^n y_i^3 & \sum_{i=0}^n y_i^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^n y_i x_i \\ \sum_{i=0}^n y_i^2 x_i \\ \sum_{i=0}^n y_i^3 x_i \end{bmatrix} \quad (3.10)$$

A obtenção dos parâmetros da trajetória é feita através da resolução do sistema matricial da Equação 3.10. Com os elementos do vetor A bem definidos, é possível determinar, para cada valor de altura na imagem (y), um ponto correspondente da trajetória (x). Desta forma, um determinado frame é extraído (Figura 15 - Etapa 1), passa pelo YOLOv8-seg (Figura 15 - Etapa 2) para a obtenção da máscara binária (Figura 15 - Etapa 3). Assim, é possível obter um vetor de coordenadas $B(x, y)$ (x é a coordenada longitudinal e y é a coordenada latitudinal da imagem) (Figura 15 - Etapa 4) que passa pelo regressor OLS (Figura 15 - Etapa 5) que, por fim, determina os parâmetros para a estimativa da trajetória (Figura 15 - Etapa 6). A Figura 15 resume as etapas apresentadas até o momento, desde a aquisição da imagem até a estimativa paramétrica.

Figura 15 – Passos para obtenção dos parâmetros da trajetória para abordagem IBVS.



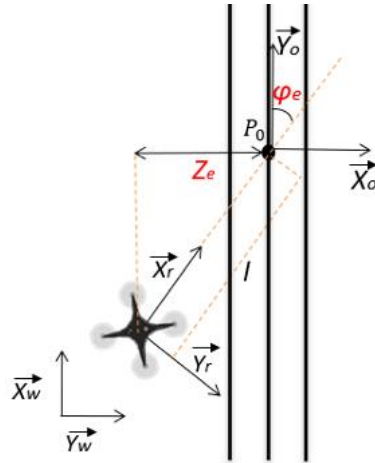
Fonte: Elaborado pelo autor.

3.4 Cálculo das variáveis de erros visuais

Com a trajetória a ser seguida bem definida, é possível determinar os erros de estado da planta que servirão de parâmetros para a escolha dos ganhos do controlador e das ações que deverão ser tomadas para manter o drone próximo do referencial escolhido. As linhas de distribuição podem ocupar uma grande porção da imagem, dependendo de fatores como a disposição dos cabos e a altitude do drone. Assim, um referencial adequado é o centro da imagem, o que permite que as linhas sejam observáveis em sua totalidade na maior parte da inspeção, independentemente de fatores como o vento, que pode alterar a posição do drone e ocasionar a perda do referencial. Além disso, espera-se que o drone se oriente paralelamente aos cabos durante a missão. Desta forma, duas variáveis de erro podem ser inferidas: Z_e , que representa a distância do ponto central do objeto ao referencial adotado, e φ_e , que indica a diferença de orientação entre o vetor latitudinal do drone e o vetor latitudinal do objeto. A Figura 16 mostra a representação aérea do problema de alinhamento e seguimento abordado neste trabalho. Os vetores \vec{X}_w , \vec{Y}_w e \vec{Z}_w estão relacionados ao sistema de coordenadas do mundo W , enquanto \vec{X}_r , \vec{Y}_r e \vec{Z}_r estão relacionados com o sistema de coordenadas do robô R , e \vec{X}_o , \vec{Y}_o referem-se ao plano

do objeto. l é a distância entre a projeção do ponto central do drone no plano do objeto e a origem do sistema de coordenadas P_0 que coincide com a posição central do objeto.

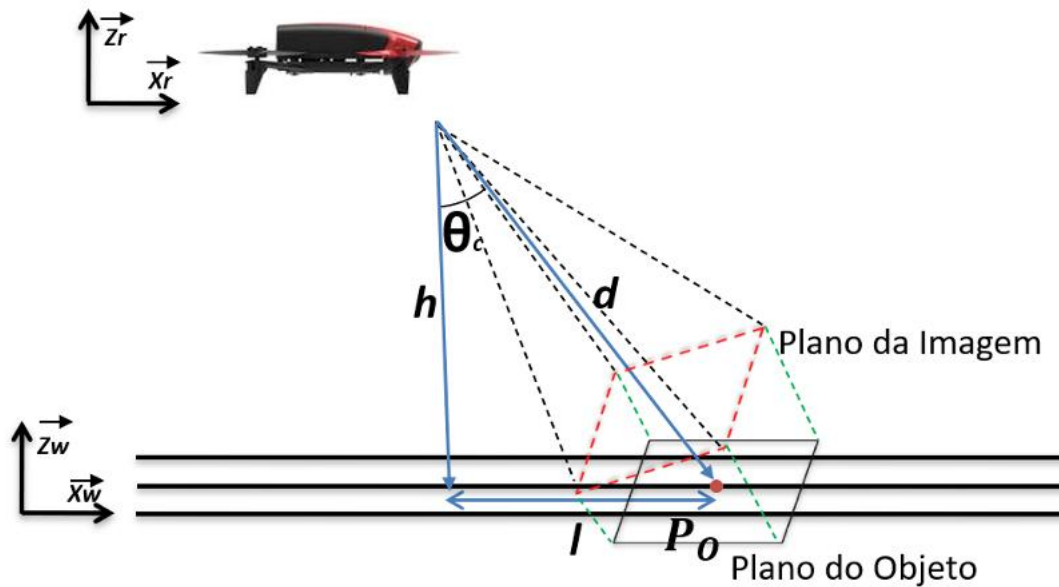
Figura 16 – Visão aérea do problema abordado neste trabalho



Fonte: Elaborado pelo autor.

A visão lateral do problema proposto é mostrada na Figura 17. h é a altitude relativa do drone em relação às linhas, θ_c é a inclinação do eixo ortogonal da imagem com o eixo vertical do drone e d é a distância da câmera ao ponto P_0 .

Figura 17 – Visão lateral do problema abordado neste trabalho



Fonte: Elaborado pelo autor.

A partir de relações geométricas, é possível inferir que:

$$l = h \cdot \tan(\theta_c) \quad (3.11)$$

$$d = \frac{h}{\cos(\theta_c)} \quad (3.12)$$

Observando a Figura 17, nota-se que, como θ_c não é nulo, o plano da imagem não é coincidente com o plano do objeto observado. Essa não coincidência cria distorções na estimativa da posição e orientação dos cabos, uma vez que um conjunto de coordenadas contidas no plano da imagem não corresponde às mesmas coordenadas no plano que é utilizado como referencial para a movimentação do drone. Assim, é necessária a aplicação de uma transformação que rotacione a imagem de forma que a perspectiva utilizada para o cálculo dos erros de trajetória seja coincidente com o plano do objeto a ser seguido. Este processo é conhecido

como Homografia, apresentado com mais detalhes em [Szeliski 2022]. A seguir, serão abordados os pontos principais aplicados neste trabalho.

Seja \tilde{x}_0 , um vetor de coordenadas bidimensionais de uma imagem em um determinado plano, composto pelos elementos $(x_0, y_0, 1)$. Uma transformação de perspectiva é dada pela seguinte relação:

$$\tilde{x}_1 = H\tilde{x}_0 \quad (3.13)$$

Em que \tilde{x}_1 é o vetor de coordenadas da imagem transformada e H é a matriz de homografia. Para a aplicação da rotação de uma imagem, H pode ser definido como:

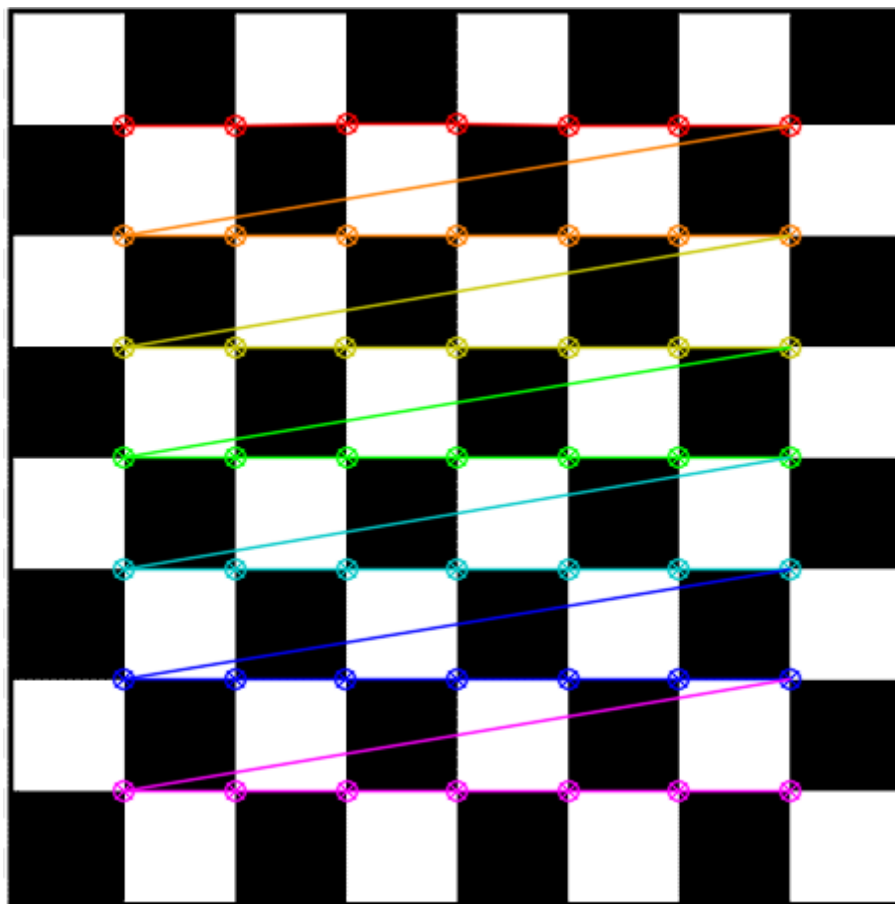
$$H = KRK^{-1} \quad (3.14)$$

K é uma matriz de calibração que está relacionada aos parâmetros intrínsecos da câmera, dada por:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (3.15)$$

Onde f_x, f_y são as distâncias focais e c_x, c_y são os centros ópticos da imagem. Embora esses parâmetros estejam relacionados ao modelo da câmera utilizado, é possível realizar uma calibração manual, a fim de obter valores mais precisos para a aplicação desejada. Este processo pode ser feito por um conjunto de funções nativas do pacote de código aberto OPENCV, que visam encontrar as coordenadas 2D das bordas de uma imagem com padrões quadriculados de dimensões conhecidas, como, por exemplo, um tabuleiro de xadrez, e associá-las a pontos 3D relacionados à localização dessas bordas no sistema de coordenadas global. O referencial teórico da metodologia é abordado com mais detalhes por Zhang [Zhang 2002]. A Figura 18 mostra um exemplo do processo de identificação das coordenadas 2D realizado durante o processo de calibração da câmera. Os pontos coloridos indicam as bordas internas do tabuleiro, identificadas pelo algoritmo de calibração.

Figura 18 – Processo de calibração da câmera para correção de perspectiva da imagem.



Fonte: Elaborado pelo autor.

A matriz R da Equação 3.14 refere-se a uma matriz de rotação que, como o nome sugere, aplica rotações nos eixos de coordenadas da imagem. Ela é composta pela multiplicação de três matrizes: R_x que aplica uma rotação em torno do eixo X de ângulo ϕ , R_y que aplica uma rotação em torno do eixo Y de ângulo θ e R_z que aplica uma rotação em torno do eixo Z de ângulo ψ . Os ângulos de rotação ϕ , θ , ψ são conhecidos como ângulos de Euler. É importante salientar que, neste trabalho, foi aplicada uma rotação em torno do eixo Y, de acordo com a inclinação da câmera θ_c . Os demais eixos não foram alterados; porém, estes serão representados de forma a manter a generalidade do equacionamento. Desta forma, reduzindo os termos

sen por s e cos por c , tem-se que:

$$R_x = \begin{bmatrix} c(\phi) & s(\phi) & 0 \\ -s(\phi) & c(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.16)$$

$$R_y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\theta) & s(\theta) \\ 0 & -s(\theta) & c(\theta) \end{bmatrix} \quad (3.17)$$

$$R_z = \begin{bmatrix} c(\psi) & s(\theta) & 0 \\ -s(\theta) & c(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.18)$$

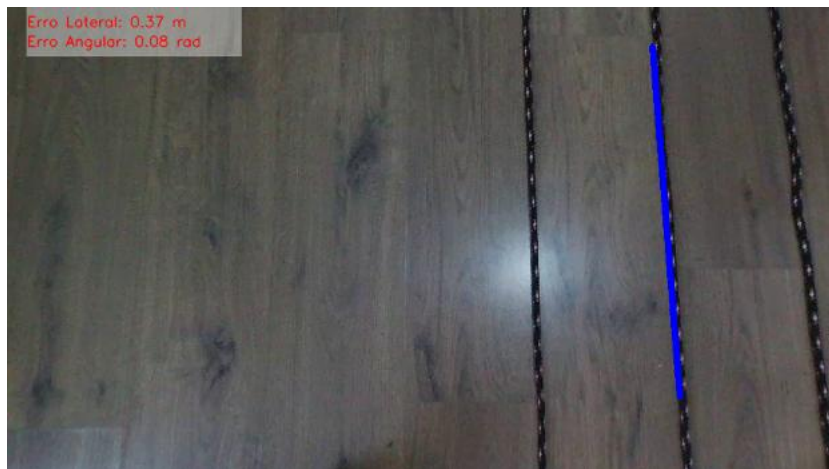
$$R = R_x R_y R_z \quad (3.19)$$

$$R = \begin{bmatrix} c(\phi)c(\psi) - s(\psi)c(\theta)s(\phi) & s(\phi)c(\psi) + c(\psi)c(\theta)s(\phi) & s(\theta)s(\phi) \\ -c(\phi)s(\psi) - s(\psi)c(\theta)c(\phi) & -s(\phi)s(\psi) + c(\psi)c(\theta)c(\phi) & s(\theta)c(\phi) \\ s(\psi)s(\theta) & -c(\psi)s(\theta) & c(\theta) \end{bmatrix} \quad (3.20)$$

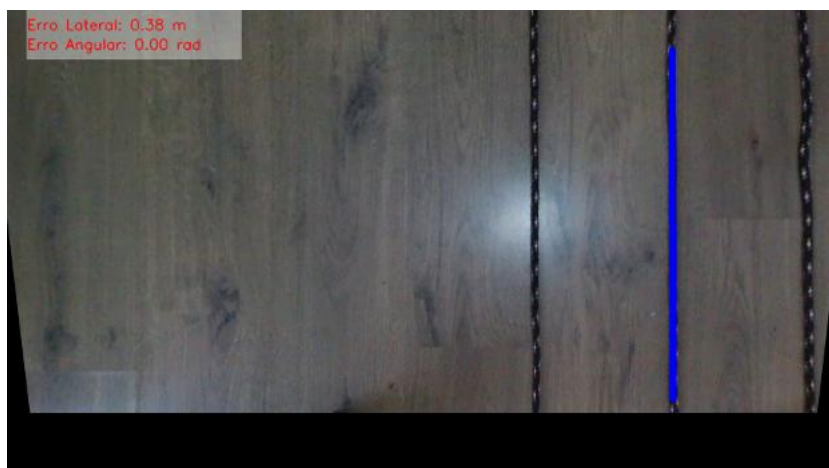
Com as matrizes R e K bem definidas, a homografia da imagem pode ser realizada. Um exemplo prático da sua implementação pode ser visto na Figura 19, na qual a primeira imagem foi obtida antes do processo de homografia e a segunda representa o pós-retificação. As imagens foram obtidas alinhando-se o drone com as linhas, de forma que o erro angular fosse nulo; ou seja, deve haver paralelismo entre as linhas e o eixo Y da imagem. É possível observar que a retificação corrigiu a distorção angular que existia devido à disparidade entre o plano da imagem e o plano do objeto, permitindo que o alinhamento seja preciso, independente da posição das linhas no campo de visão da câmera.

Figura 19 – Exemplo da aplicação da Homografia na imagem do drone.

(a) Imagem pré-homografia



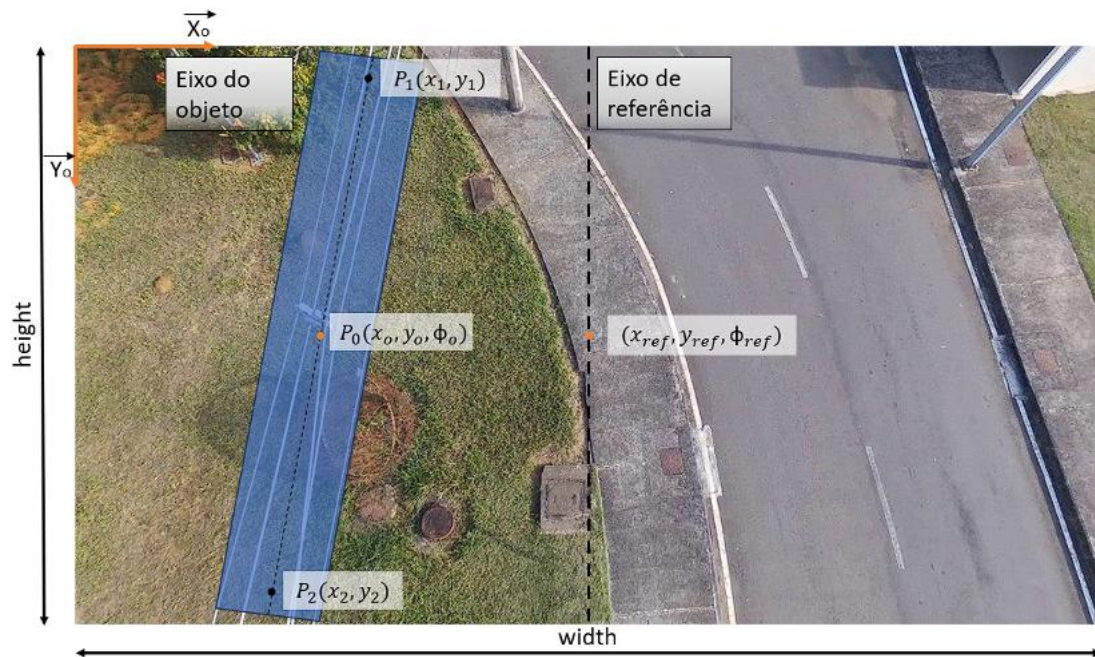
(b) Imagem pós-homografia.



Fonte: Elaborado pelo autor.

Feita a retificação da imagem, o próximo passo é traduzir as representações gerais do problema apresentadas pelas Figuras 16 e 17 em uma abordagem que permita o controle do drone por meio de informações visuais. A Figura 20 apresenta a ideia geral do esquema de controle empregado diretamente no plano do objeto. A partir da trajetória, que pode ser determinada através dos passos apresentados na seção 3.3, é possível obter as coordenadas do ponto P_0 , que é utilizado para a

Figura 20 – Estimativa dos erros de posição e de alinhamento com base na abordagem IBVS.



Fonte: Elaborado pelo autor.

determinação da posição lateral, e dos pontos adicionais P_1 e P_2 , que são utilizados para a determinação da orientação das linhas a serem seguidas.

Os pontos de referência $x_{ref}, y_{ref}, \phi_{ref}$ podem ser representados pelas equações:

$$x_{ref} = \frac{w}{2} \quad (3.21)$$

$$y_{ref} = \frac{h}{2} \quad (3.22)$$

$$\phi_{ref} = 0 \quad (3.23)$$

Em que w e h representam, respectivamente, o comprimento e a largura da imagem, em *pixels*. Desta forma, é possível determinar as variáveis Δ_x e Δ_ϕ ,

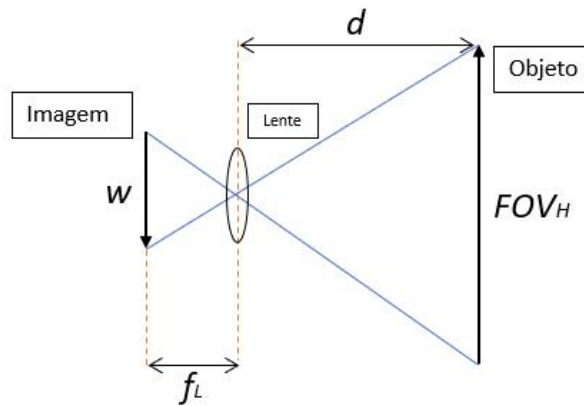
que indicam o quão distante e desalinhado o drone está da referência adotada, por meio das equações:

$$\Delta_x = x_0 - x_{ref} = x_0 - \frac{w}{2} \quad (3.24)$$

$$\Delta_\phi = \phi_0 - \phi_{ref} = \tan^{-1} \left(\frac{y_1 - y_2}{x_1 - x_2} \right) \quad (3.25)$$

O próximo passo é transformar as variáveis Δ_x e Δ_ϕ , que são representadas por unidades de *pixels*, em variáveis de erro que são adotadas através de unidades métricas (metros) e angulares (radianos). Com isso, é possível definir o erro angular φ_e como a própria variação angular Δ_ϕ . Já para o erro lateral Z_e , é necessário transformar Δ_x por meio de um fator de conversão k que leva em conta parâmetros como a distância focal da câmera (f_L), o campo de visão horizontal da câmera FOV_H e a distância da lente para o objeto. As relações geométricas necessárias para a computação do fator k são apresentadas na Figura 21.

Figura 21 – Extração de parâmetros visuais através de uma imagem



Fonte: Elaborado pelo autor.

A partir da Figura 21, é possível extrair a seguinte relação:

$$\frac{FOV_H}{w} = \frac{d}{f_L} = k \quad (3.26)$$

A variável d pode ser obtida através da Equação 3.12. Assim:

$$k = \frac{h}{\cos(\theta_c)f_L} \quad (3.27)$$

Por fim, a variável de erro lateral Z_e pode ser determinada da seguinte forma:

$$Z_e = k.\Delta_x = \frac{h.(x_0 - \frac{w}{2})}{\cos(\theta_c)f_L} \quad (3.28)$$

Tanto Z_e quanto φ_e são inseridos na malha de controle, que tem como objetivo minimizá-los por meio da aplicação de velocidades lineares e angulares que serão adotadas pelo drone durante as missões de inspeção. Os pontos relacionados à estrutura de controle utilizada neste trabalho serão abordados na seção 3.5.

3.5 Controle de movimentação

Os drones multirotores, como os utilizados neste trabalho, constituem uma classe específica de VANT cuja movimentação é baseada no empuxo gerado pelas hélices acopladas aos motores. O controle da rotação desses rotores é realizado por meio da variação da frequência da corrente elétrica aplicada aos enrolamentos dos motores. Para executar um determinado movimento, como decolagem ou guinada, o drone utiliza os sensores embarcados para estimar seu estado atual, incluindo velocidades, posições e orientações, e, a partir disso, determinar a velocidade de rotação de cada hélice.

Nesse contexto, o controle do veículo pode ser descrito por meio de quatro sinais principais: três velocidades lineares, v_x , v_y e v_z , associadas aos deslocamentos ao longo dos eixos do referencial do corpo X_r , Y_r e Z_r , respectivamente, e uma velocidade angular ω_z , associada à rotação em torno do eixo Z_r .

Adicionalmente, a orientação do drone é descrita por meio dos ângulos de atitude, que representam rotações em torno dos eixos do corpo: arfagem (pitch), rolagem (roll) e guinada (yaw). Esses ângulos não devem ser confundidos com as velocidades lineares, uma vez que descrevem exclusivamente o estado angular do veículo.

A Figura 22 apresenta a definição dos eixos do corpo do drone, utilizados como referencial para a aplicação dos comandos de controle.

Figura 22 – Representação dos eixos do drone utilizado neste trabalho



Fonte: Elaborado pelo autor.

A abordagem de controle IBVS proposta neste trabalho opera da seguinte forma: as informações visuais obtidas pela câmera do drone são transmitidas a um computador portátil, responsável pelas etapas de identificação dos cabos, estimação da trajetória e cálculo dos erros de estado previamente definidos. Em seguida, o nó de controle utiliza esses erros para determinar as velocidades lineares v_y e a velocidade angular ω_z , de modo a manter o objeto de interesse alinhado ao referencial adotado.

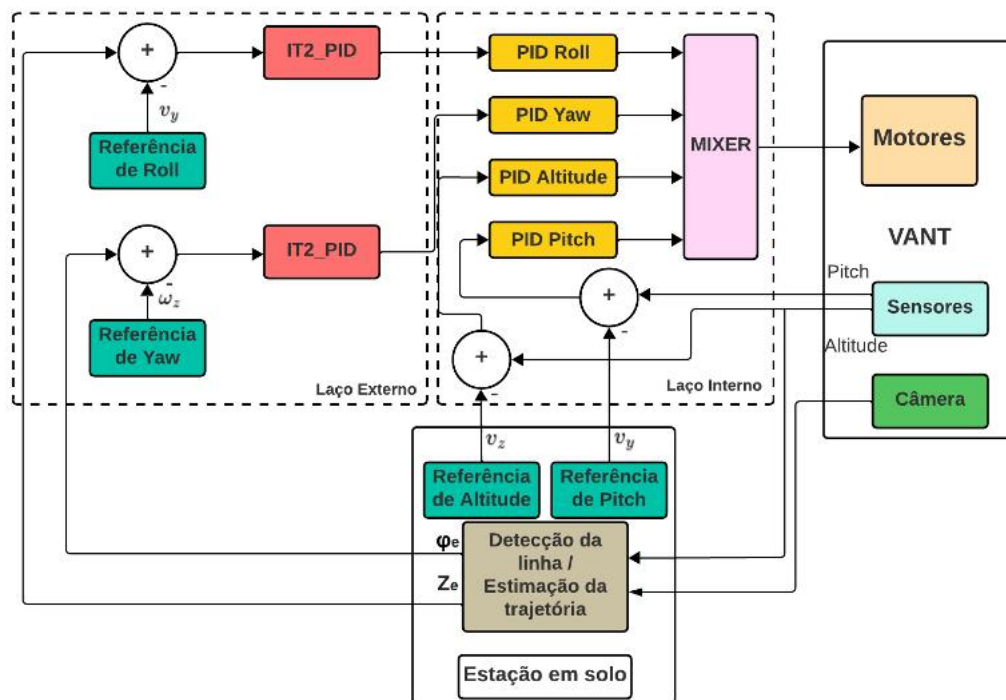
A velocidade v_x é definida de acordo com a missão, sendo nula em tarefas de alinhamento angular e lateral, ou assumindo um valor constante predefinido em missões de seguimento. A velocidade v_z , por sua vez, é utilizada para manter o drone em altitude aproximadamente constante durante a execução da missão, não sendo necessário, portanto, um controlador externo dedicado para essas variáveis.

No nível embarcado, o drone utiliza dados provenientes de acelerômetros, magnetômetros e giroscópios para estimar seu estado atual. Esses dados são combinados com as referências de velocidade fornecidas pelo sistema externo e enviados ao bloco mixer, responsável por calcular os sinais de PWM aplicados a cada motor, de forma a viabilizar a execução desacoplada dos movimentos.

A Figura 23 apresenta a estrutura de controle desenvolvida neste trabalho. O laço interno é executado embarcado no drone, utilizando diretamente os dados dos sensores e da câmera, os quais também são transmitidos à estação em solo, responsável pelo laço externo. O kit de desenvolvimento Parrot ARSDK3 gerencia a aquisição dos dados de telemetria, a transmissão do fluxo de imagens e a distribuição dos sinais de controle necessários à movimentação do Bebop 2. A comunicação com o computador portátil é realizada por meio do driver `bebop_autonomy`, utilizando a plataforma ROS, que também é responsável pelo gerenciamento das missões e pelo envio dos comandos de velocidade. No ROS, essas operações são implementadas por meio de mecanismos de publicação e subscrição em tópicos, executados de forma síncrona. A conexão entre o drone e a estação em solo é estabelecida via rede Wi-Fi.

Ressalta-se que as referências associadas às velocidades v_x e v_z são definidas como valores constantes ao longo da missão, garantindo, respectivamente, o avanço ao longo da trajetória e a manutenção da altitude desejada. Por outro lado, as velocidades v_y e ω_z são ajustadas dinamicamente pelo controlador IBVS, com o objetivo de manter o drone alinhado lateralmente e angularmente em relação às linhas de distribuição.

Figura 23 – Representação dos eixos do drone utilizado neste trabalho



Fonte: Elaborado pelo autor.

3.5.1 Controle adaptativo IT2_PID

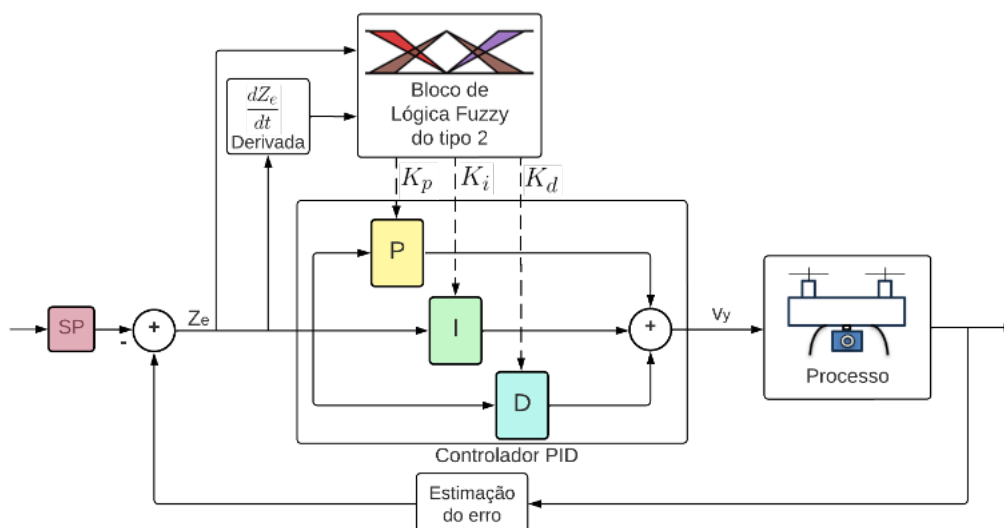
Conforme visto na Figura 23, os blocos IT2_PID são responsáveis pelo controle dos movimentos de *roll* e *pitch* do drone, que, na prática, estão relacionados às tarefas de alinhamento lateral e angular, respectivamente. O princípio de funcionamento deste controlador pode ser entendido em duas etapas. Inicialmente, a posição e a orientação relativas do drone em relação às linhas são estimadas e, em seguida, comparadas com as referências (setpoints ou SP), resultando em valores de erro. Essas variáveis, em conjunto com suas derivadas, servem como entrada para o bloco de lógica fuzzy do tipo 2.

Internamente, esse bloco transforma as variáveis precisas de entrada em números fuzzy por meio de funções de pertinência, processando-as com base em um conjunto de regras que relaciona diferentes níveis de erro e sua variação a diferentes

níveis de ganhos, os quais são posteriormente convertidos em valores exatos para utilização no controlador PID, ou seja, os ganhos são atualizados periodicamente. Na segunda etapa, os ganhos adaptativos são combinados com o erro e inseridos nos blocos proporcional, integral e derivativo, sendo a saída do controlador obtida pelo somatório das contribuições de cada termo. Essa saída é então aplicada à planta, gerando novos valores de posição que retroalimentam o sistema e dão continuidade ao processo.

A Figura 24 apresenta o esquemático do bloco IT2_PID desenvolvido para o controle do erro lateral, associado ao movimento de roll. O bloco referente ao controle angular, associado ao yaw, é construído de maneira similar.

Figura 24 – Representação do bloco IT2_PID utilizado neste trabalho.



Fonte: Elaborado pelo autor.

O conjunto de regras utilizado para a inferência dos valores de saída é apresentado na Tabela 3. Os erros são tratados como pequenos, médios e grandes, enquanto a variação dos mesmos assume valores negativos (erro decrescente), zero (erro mantendo-se constante) e positivos (erro crescente). A saída é representada por um conjunto de três variáveis que indicam os ganhos proporcional K_P , derivativo K_D , e α , que é um fator utilizado para o cálculo do ganho integrativo K_I . Essas

Tabela 3 – Conjunto de regras *fuzzy* para adaptação dos ganhos do bloco PID.

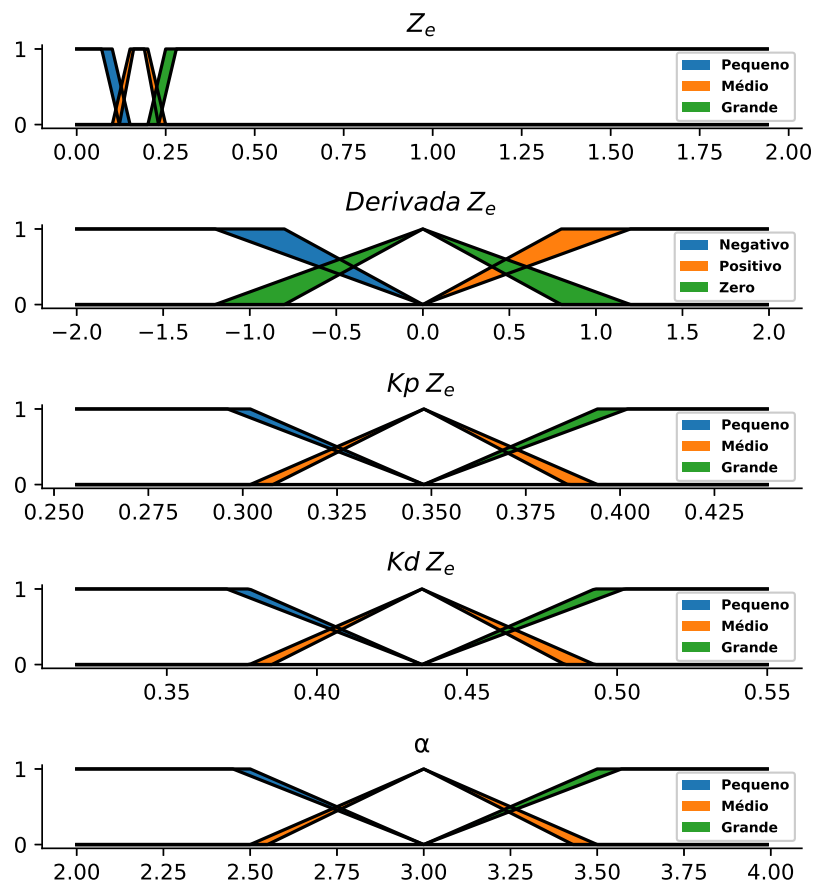
dZ_e/dt	Z_e		
	Pequeno	Médio	Grande
Negativo	P/G/G	M/M/M	G/P/P
Zero	G/G/M	G/M/P	G/P/P
Positivo	P/G/G	M/M/M	G/P/P

variáveis podem assumir valores pequenos (P), médios (M) ou grandes (G). A base de regras foi elaborada de acordo com os trabalhos de [Andrade et al. 2021], que propôs um controlador T1-PID para o controle de movimento de um drone generalizado. A combinação de três estados para o erro e sua variação propicia nove possibilidades de estado para os ganhos, que variam de acordo com o comportamento do processo em determinado instante. O ganho integral é obtido através do fator α , por meio de uma relação entre os ganhos proporcionais e derivativos, de acordo com a seguinte equação:

$$K_I = \frac{K_p^2}{\alpha K_d} \quad (3.29)$$

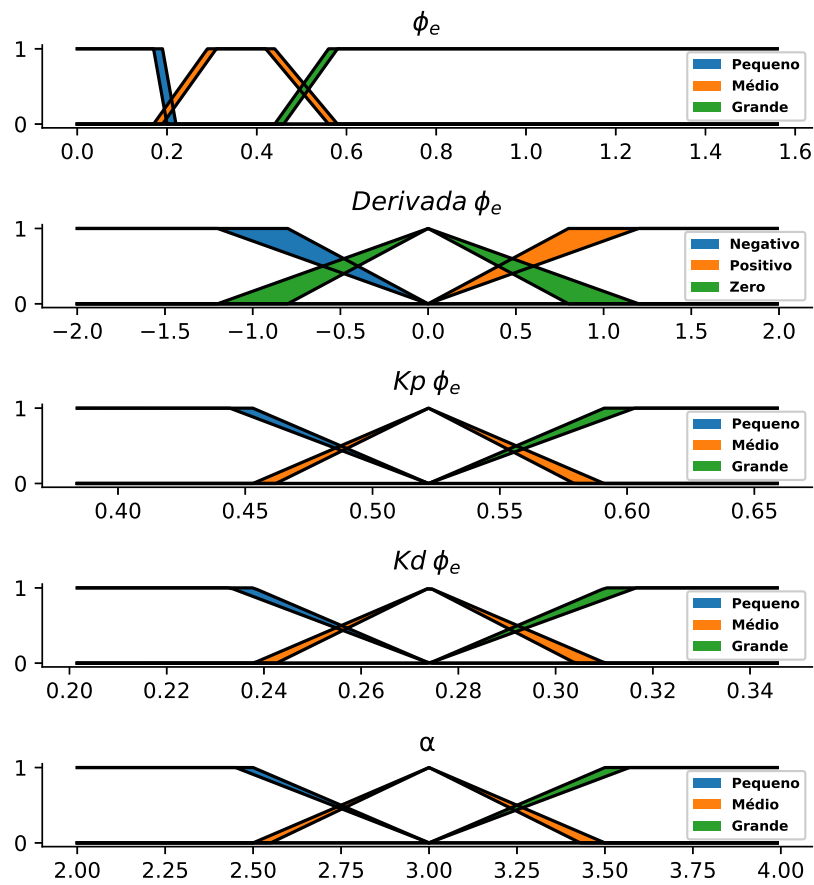
As funções de pertinência adotadas neste trabalho podem ser vistas nas Figuras 25 e 26. As notações $IT2_PID_{Z_e}$ e $IT2_PID_{\varphi_e}$ foram adotadas para a diferenciação das funções utilizadas para o controlador lateral e angular, respectivamente.

Figura 25 – Funções de pertinência para o bloco $IT2_PID_{Z_e}$.



Fonte: Elaborado pelo autor.

Figura 26 – Funções de pertinência para o bloco $IT2_PID_{\phi_e}$.



Fonte: Elaborado pelo autor.

Com base na análise gráfica das figuras acima, observa-se que tanto as variáveis de entrada quanto as de saída do bloco fuzzy estão restritas a uma faixa definida de valores mínimos e máximos. Para o erro e sua variação, foram considerados os limites do campo visual da câmera, bem como a altitude relativa do drone em relação às linhas, além de valores plausíveis para a derivada temporal dos erros.

Quanto aos ganhos do controlador, a faixa utilizada foi determinada com base na metodologia proposta inicialmente por [Larson 2016] e, posteriormente, aprimorada por [Andrade et al. 2021]. Primeiramente, realizam-se testes empíricos

nos quais os ganhos derivativo e integrativo são fixados em zero, enquanto o ganho proporcional é aumentado progressivamente até que o erro apresente oscilações consistentes. Com isso, obtém-se o valor do ganho final, K_u , e o período de oscilação do sistema T_u . A partir desses parâmetros, os limiares intermediários são definidos por meio das seguintes equações [Andrade et al. 2021]:

$$K_{P',min} = 0.32K_u \quad (3.30)$$

$$K_{D',min} = 0.08K_uP_u \quad (3.31)$$

$$K_{P',max} = 0.6K_u \quad (3.32)$$

$$K_{D',max} = 0.15K_uP_u \quad (3.33)$$

Os valores limites dos ganhos passam por um processo de compressão, o que permite que o drone realize os movimentos de maneira sútil, evitando movimentações bruscas que podem danificar o veículo e afetar a qualidade das imagens. Assim, os limiares definitivos são calculados através das equações:

$$K_{P,min} = K_{P',min} - \frac{K_{P',min}}{5} \quad (3.34)$$

$$K_{D,min} = K_{D',min} - \frac{K_{D',min}}{5} \quad (3.35)$$

$$K_{P,max} = K_{P',min} + \frac{K_{P',max}}{5} \quad (3.36)$$

$$K_{D,max} = K_{D',min} + \frac{K_{D',max}}{5} \quad (3.37)$$

Para o cálculo de K_u e T_u , foram realizados experimentos em um ambiente controlado, no qual o Bebop 2 sobrevoou cordas dispostas de maneira semelhante a cabos de distribuição a uma altura compatível com o cenário real. Com isso, para o controlador lateral $IT2_PID_{Z_e}$, foram obtidos os valores $K_u = 1,5$ e $T_u = 2,1s$, enquanto que para o controlador angular, os valores encontrados foram $K_u = 1$ e

$T_u = 5s$. Essa metodologia é adaptável a diferentes modelos de drone em diversos tipos de aplicações, sendo esse um dos motivos para sua escolha neste trabalho.

3.6 Conclusões Parciais

O presente capítulo apresentou uma visão detalhada da metodologia proposta para a inspeção das linhas de distribuição de energia, cobrindo todas as etapas, desde a escolha do tipo de missão a ser executada até o cálculo dos sinais de controle que são atribuídos ao drone durante a missão. Internamente, foram abordadas todas as questões essenciais para o funcionamento do que foi proposto, como a estimação da trajetória e, por consequência, os erros visuais que são atribuídos aos controladores, a estrutura de controle completa com os laços internos e externos que funcionam em conjunto para manter o drone em uma certa altitude, enquanto se mantém alinhado com os cabos. Além disso, foram apresentadas as questões inerentes aos blocos de lógica *fuzzy*, como o conjunto de regras e as funções de pertinência empregadas para a adaptação dos ganhos.

Durante o capítulo, também foram abordados tópicos secundários, como o processo de construção do banco de imagens para a detecção das linhas e dos outros elementos do sistema de distribuição, o treinamento do YOLOv8 de maneira customizada, atendendo às necessidades do trabalho, a técnica utilizada para a determinação da faixa na qual os ganhos do controlador podem variar, além da comunicação entre o drone e a estação em solo que gerencia a missão e é responsável pela determinação e pelo envio dos sinais necessários para o controle do drone. No próximo capítulo, serão apresentados os resultados que corroboram os tópicos apresentados nesta seção, tanto em cenário simulado, quanto em ambientes reais.

4 Resultados e Discussões

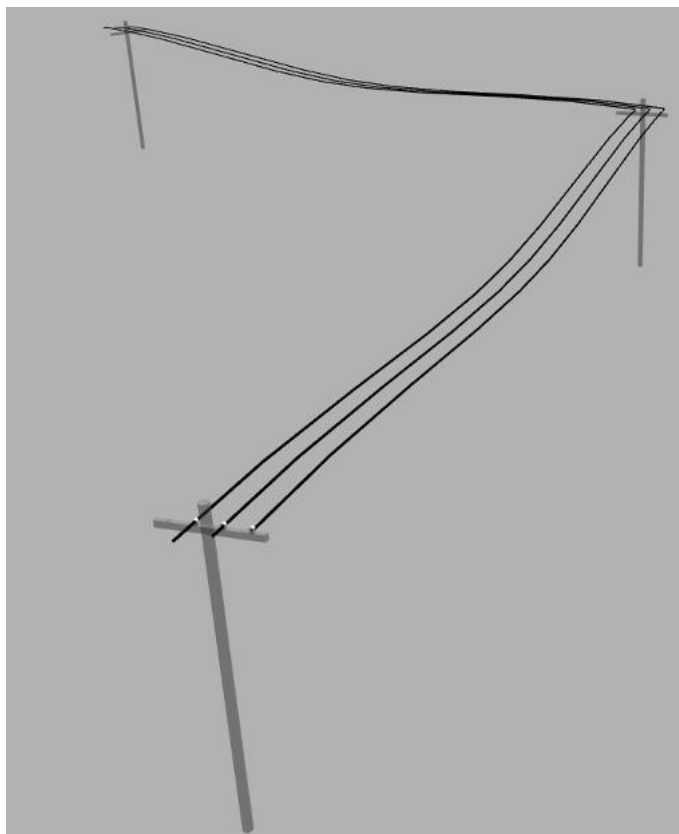
O Capítulo 4 apresenta os resultados obtidos que validam a metodologia proposta. A Seção 4.1 refere-se aos resultados obtidos por meio de simulações através do simulador Gazebo. Nesta etapa, o foco foi apresentar a eficácia do controlador IT2_PID para o problema de alinhamento e seguimento de linhas de distribuição, comparando este controlador híbrido com um controle similar utilizando lógica *fuzzy* do tipo 1 e controladores PID de ganhos fixos. Foram realizados testes envolvendo as missões de alinhamento lateral e angular, o efeito da mudança do conjunto de regras no comportamento do sistema e, por fim, uma simulação de uma missão de inspeção de linhas. Na seção 4.2 são realizados testes envolvendo cenários reais que contêm linhas de distribuição. Inicialmente, foram realizados testes para comprovar a eficácia e viabilidade da utilização do YOLOv8-seg para identificação dos cabos de energia e do YOLOv8 para detecção de elementos presentes no sistema de distribuição em uma aplicação em regime de baixa latência. Em seguida, a metodologia completa foi testada por meio do drone Bebop 2 em uma linha de distribuição real para as tarefas de alinhamento, seguimento e inspeção.

4.1 Resultados em ambiente simulado

Antes que a metodologia completa possa ser testada em cenários reais, é necessário confirmar a eficácia do funcionamento do bloco de controle em um ambiente ideal por meio de simulações computacionais. Assim, qualquer eventual erro na implementação pode ser corrigido sem prejuízos resultantes da perda de componentes ou de acidentes causados por ações inesperadas do drone. Pensando nisso, foi escolhido o Gazebo, um simulador de código aberto capaz de integrar sensores e atuadores de sistemas robóticos, como um drone, com modelos tridimensionais que podem ser construídos por meio de softwares como Blender e Solidworks. Assim, foi criado um cenário composto por torres de distribuição de energia ligadas por cabos elétricos, que pode ser visto na Figura 27. O drone é simulado por meio do PX4, um controlador de voo que permite simulações por meio da subscrição e publicação de tópicos, sendo possível a integração com a plataforma ROS. Essa

configuração pode ser facilmente transcrita para drones reais, sendo esse o principal motivo da sua escolha para os testes iniciais.

Figura 27 – Cenário criado no Gazebo para realização dos testes em ambiente simulado.



Fonte: Elaborado pelo autor.

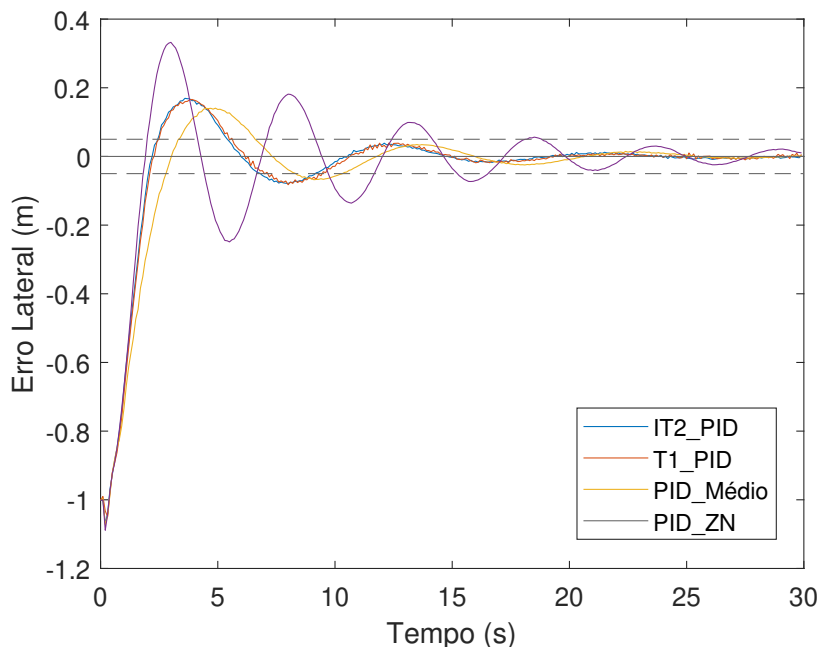
Todos os testes envolvendo simulações foram realizados em uma máquina com um processador Intel Core i5-10400F, uma placa gráfica AMD Radeon RX 5600 XT e 16 GB de RAM. O sistema operacional escolhido é o Ubuntu 20.04.5 LTS 64-bits com ROS Noetic.

4.1.1 Controlador Lateral

Inicialmente, foram realizados testes que permitissem verificar se o controlador lateral é capaz de levar o drone à posição especificada acima dos cabos de energia. Para esse fim, o drone foi posicionado quatro metros acima das linhas, a uma distância lateral de um metro em relação ao cabo central. O objetivo final é manter o drone acima das linhas, tendo o centro da imagem como referencial. O controlador proposto neste trabalho foi comparado a um controle semelhante, utilizando a lógica *fuzzy* do tipo 1, com o mesmo perfil das funções de pertinência e conjunto de regras, além de controles PID de ganhos fixos; o primeiro sintonizado através da técnica de Ziegler-Nichols, e o segundo adotando os ganhos médios calculados pelas abordagens *fuzzy*. O objetivo destes testes é verificar se existem melhorias significativas ao utilizar a lógica *fuzzy* do tipo 2 para a escolha dos ganhos e se a adaptabilidade desses ganhos de acordo com os estados das variáveis de erro afeta o desempenho global do controle.

O perfil temporal médio do erro lateral após o início do alinhamento pode ser visto na Figura 28. É importante salientar que todos os testes envolvendo os cenários simulados foram repetidos 50 vezes e, sendo assim, as análises gráficas são representadas pelo valor médio de todas as simulações. As linhas pontilhadas representam os limiares do estado estacionário do sistema.

Figura 28 – Comparativo da magnitude do erro lateral para os quatro controladores abordados



Fonte: Elaborado pelo autor.

A Tabela 4 apresenta os dados quantitativos referentes às métricas de controle para o erro lateral. IAE é a integral do erro absoluto, M_p é o pico máximo, T_p é o instante de pico, T_s é o tempo de estabilização e E_{ss} é o erro de estado estacionário. Além disso, o tempo de processamento computacional para cada controlador foi calculado.

Tabela 4 – Resultados quantitativos para o controle lateral em cenário simulado

Controlador	IAE	$M_p(m)$	$T_p(s)$	$T_s(s)$	$E_{ss}(m)$	Tempo de processamento (s)
IT2_PID	2.3392 ± 0.1869	0.2005 ± 0.0572	3.72 ± 0.3423	9.21 ± 0.48	0.0164 ± 0.0078	0.0103 ± 2.57 × 10 ⁻⁴
T1_PID	2.4742 ± 0.1896	0.2111 ± 0.0545	3.75 ± 0.3823	9.45 ± 0.54	0.0209 ± 0.0090	0.0077 ± 1.7267 × 10 ⁻⁴
PID_Médio	2.5490 ± 0.1746	0.1563 ± 0.0292	4.85 ± 0.3823	10.32 ± 0.61	0.0184 ± 0.0075	1.0873 × 10 ⁻⁴ ± 1.9944 × 10 ⁻⁵
PID_ZN	3.3122 ± 0.1634	0.3437 ± 0.0252	3.05 ± 0.1526	18.91 ± 0.46	0.0219 ± 0.0057	1.0310 × 10⁻⁴ ± 1.3417 × 10⁻⁵

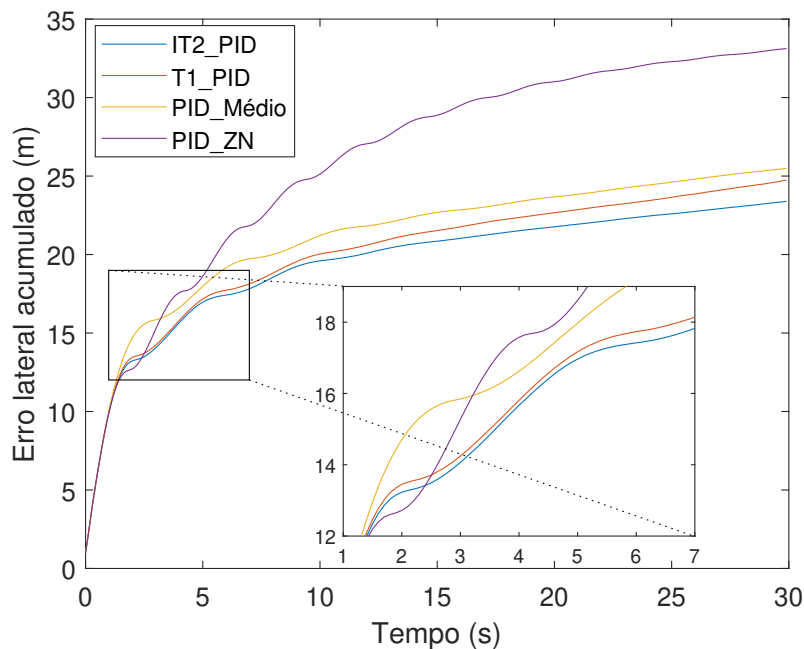
Por meio da análise gráfica da Figura 28, é possível observar que os quatro controladores foram capazes de levar o drone ao valor de referência adotado. No entanto, o controlador PID sintonizado por Ziegler-Nichols apresentou oscilações

longas e duradouras, o que não é desejável em missões nas quais um determinado objeto precisa ser observado. Os dados quantitativos indicam que os controladores de ganhos adaptáveis apresentaram os menores valores para a métrica IAE, que computa o erro total do modelo, quando comparados com as abordagens de ganhos fixos. O menor valor de pico foi obtido através do controlador PID com os ganhos médios, embora o tempo para atingir esse pico e o tempo necessário para o acomodamento do erro tenham sido maiores do que para os controladores híbridos. Este fato corrobora a capacidade da lógica *fuzzy* em lidar com sistemas que apresentam diversos tipos de comportamento. O modelo pode, por exemplo, inferir cenários em que o erro se estagnou e forçar a diminuição dele por meio do ajuste da magnitude de cada componente do bloco PID.

A abordagem IT2_PID apresentou resultados superiores em todas as métricas de controle quando comparada ao T1_PID, ao custo de um tempo de processamento levemente maior. Este aumento, porém, não afeta o funcionamento da metodologia que aplica sinais de controle ao drone em uma frequência de 10 Hz.

Para verificar o comportamento do erro ao longo de todo o período do teste, é apresentado o gráfico de erro acumulado, conforme visto na Figura 29. No início do movimento do drone, o erro acumulado manteve-se semelhante para os quatro casos. Em aproximadamente 1.5 segundos, os controladores de ganhos fixos passaram a apresentar um aumento mais significativo no erro se comparados aos controladores híbridos que, por meio da adaptação dos ganhos, conseguem propiciar um amortecimento mais contundente do movimento do drone. Aos 2 segundos, é possível observar um distanciamento da curva para as duas abordagens *fuzzy*, com a vantagem da lógica *fuzzy* do tipo 2, a qual se estende durante todo o teste, evidenciando a capacidade desta variante em lidar com sistemas com um grande grau de incertezas, tanto nos sinais de entrada quanto nos de saída.

Figura 29 – Comparativo do erro lateral acumulado para os quatro controladores abordados.



Fonte: Elaborado pelo autor.

4.1.2 Efeito da modificação do conjunto de regras

O desempenho de sistemas baseados em lógica *fuzzy* pode ser afetado por diversos fatores, sendo um deles a construção das regras que, juntamente com as funções de pertinência, participam do processo de inferência das variáveis de saída. O conjunto de regras sumarizado pela Tabela 3 serviu como base para a implementação do bloco de controle apresentado neste trabalho. Sua estrutura foi construída de forma simétrica, de modo que os ganhos do controlador se adaptam de maneira uniforme, independentemente se a derivada do erro absoluto é positiva ou negativa, ou seja, se o erro está aumentando ou diminuindo. Pensando nisso, algumas modificações foram propostas com o intuito de deixar o controlador mais agressivo em cenários nos quais o erro está aumentando e mais suave conforme o drone se aproxime da referência, propiciando um efeito de amortecimento no sistema. Além disso, o efeito da parcela derivativa foi reduzido para erros pequenos, de forma

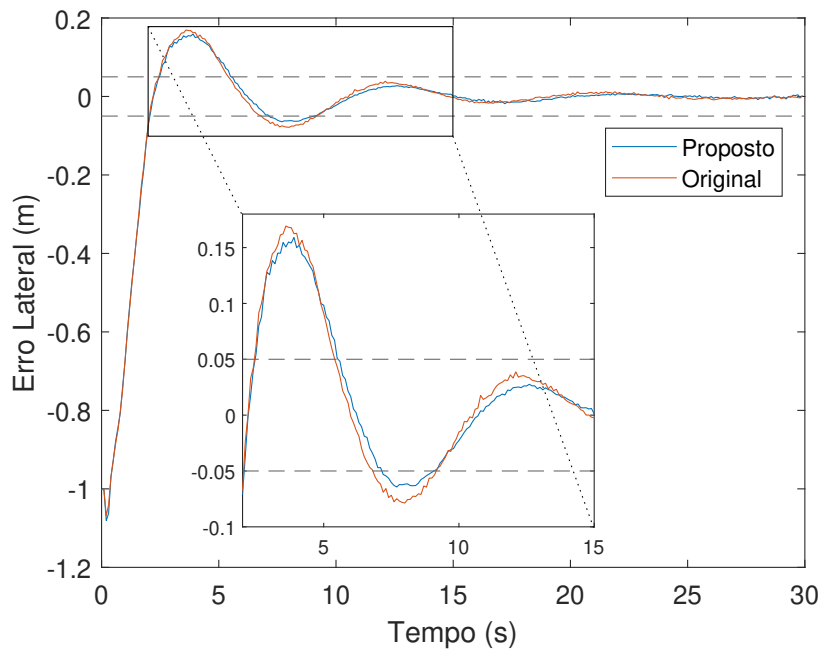
a reduzir possíveis oscilações no estado permanente causadas por inconsistências na determinação dos erros. Assim, o novo conjunto de regras proposto pode ser visto na Tabela 5, com as modificações destacadas em vermelho.

Tabela 5 – Novo conjunto de regras proposto.

dZ_e/dt	Z_e		
	Pequeno	Médio	Grande
Negativo	P/M/G	M/M/M	G/P/P
Zero	G/G/M	G/M/P	G/P/P
Positivo	M/M/G	G/M/M	G/P/P

Com isso, novos testes foram executados para a missão de alinhamento lateral. Os resultados gráficos podem ser observados na Figura 30. Os dados quantitativos são apresentados na Tabela 6.

Figura 30 – Comparativo do erro lateral médio para diferentes conjuntos de regras *fuzzy*.



Fonte: Elaborado pelo autor.

Tabela 6 – Resultados quantitativos para o controle lateral em cenário simulado.

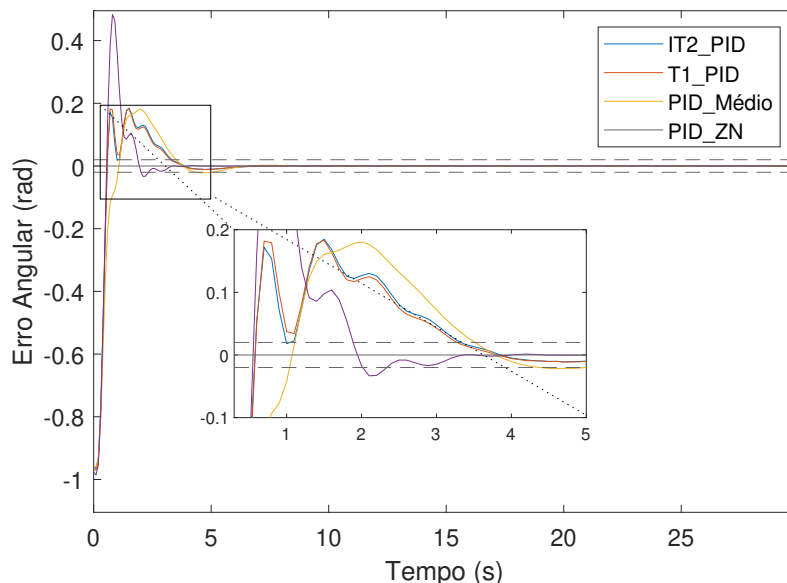
Regras	IAE	$M_p(m)$	$T_p(s)$	$T_s(s)$	$E_{ss}(m)$	Tempo de processamento (s)
Original	2.3392 ± 0.1869	0.2005 ± 0.0572	3.72 ± 0.3423	9.21 ± 0.48	0.0164 ± 0.0078	$0.0103 \pm 2.5738 \times 10^{-4}$
Proposta	2.2936 ± 0.1692	0.1887 ± 0.0407	3.99 ± 0.3821	9.11 ± 0.43	0.0170 ± 0.0070	$0.0098 \pm 3.4819 \times 10^{-4}$

Através da análise gráfica, é possível observar que a modificação do conjunto de regras propiciou uma leve redução na oscilação do drone no período transitório do movimento. Essa redução também pode ser percebida por meio da redução dos valores de M_p , o que, por consequência, também reduziu o IAE. As demais métricas não obtiveram variações significativas. Entretanto, essa análise abre um leque para novos testes envolvendo outras mudanças no conjunto de regras, bem como a modificação das funções de pertinência do modelo, de modo que o drone atinja o referencial mais rapidamente, sem grandes oscilações durante toda a missão, sendo esse um dos pontos a serem estudados em trabalhos futuros.

4.1.3 Controlador Angular

A metodologia para os testes envolvendo o alinhamento angular seguiu os mesmos padrões do alinhamento lateral. Para tal, o drone foi posicionado a 4 metros acima das linhas, com um desvio de orientação relativo de $1rad$, sendo que a quantidade e duração dos testes foram mantidas. As funções de pertinência utilizadas são as apresentadas na Figura 26, enquanto as regras adotadas são as mesmas da Tabela 5. Os resultados dos testes podem ser vistos na Figura 31 e por meio da Tabela 7.

Figura 31 – Comparativo da magnitude do erro angular para os quatro controladores abordados



Fonte: Elaborado pelo autor.

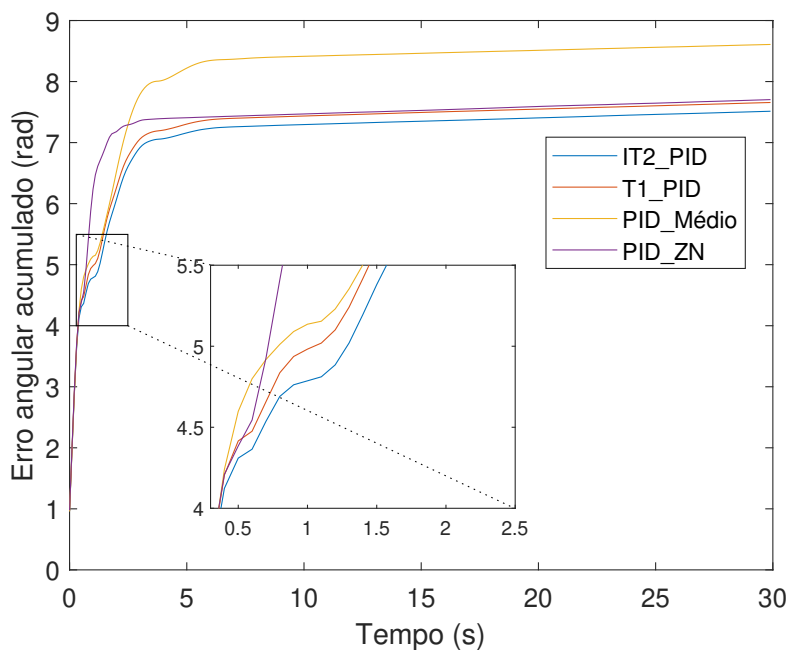
Tabela 7 – Resultados quantitativos para o controle angular em cenário simulado

Controlador	IAE	$M_p(m)$	$T_p(s)$	$T_s(s)$	$E_{ss}(m)$	Tempo de processamento (s)
IT2_PID	0.7514 ± 0.0147	0.2012 ± 0.0238	1.23 ± 0.377	3.32 ± 0.49	$1.12 \times 10^{-3} \pm 1.91 \times 10^{-4}$	$0.0150 \pm 6.1912 \times 10^{-4}$
T1_PID	0.7647 ± 0.0179	0.2194 ± 0.0555	1.22 ± 0.366	3.28 ± 0.54	$1.13 \times 10^{-3} \pm 1.77 \times 10^{-4}$	$0.0099 \pm 6.7115 \times 10^{-4}$
PID_Médio	0.8608 ± 0.0192	0.1825 ± 0.0059	2.05 ± 0.140	3.35 ± 0.71	$1.06 \times 10^{-3} \pm 1.63 \times 10^{-4}$	$1.0489 \times 10^{-4} \pm 1.5211 \times 10^{-5}$
PID_ZN	0.7704 ± 0.0262	0.4863 ± 0.0667	0.92 ± 0.043	2.34 ± 0.349	$1.18 \times 10^{-3} \pm 2.13 \times 10^{-4}$	$1.0811 \times 10^{-4} \pm 1.5917 \times 10^{-5}$

Novamente, o controlador PID_ZN apresentou um grande pico máximo evidenciado pelo alto valor de M_p . O PID_Médio mostrou um pico menor se comparado com as abordagens *fuzzy*, embora o IAE seja maior devido ao maior tempo para atingir o valor máximo. A comparação direta das duas abordagens adaptativas indica que a variante IT2_PID se sobressaiu em comparação ao T1_PID com relação ao IAE e M_p , assim como na missão de alinhamento lateral, apesar de apresentar valores levemente maiores para T_p e T_s além de um tempo de processamento mais lento. Pela análise do erro acumulado, mostrado na Figura 32, é possível inferir que a diferença entre as duas abordagens é acentuada novamente no regime transitório do sistema, mantendo-se relativamente constante durante

o estado permanente. O aumento súbito do erro acumulado para o PID_Médio evidencia a capacidade de adaptação dos ganhos conforme o comportamento do sistema muda.

Figura 32 – Comparativo do erro angular acumulado para os quatro controladores abordados.



Fonte: Elaborado pelo autor.

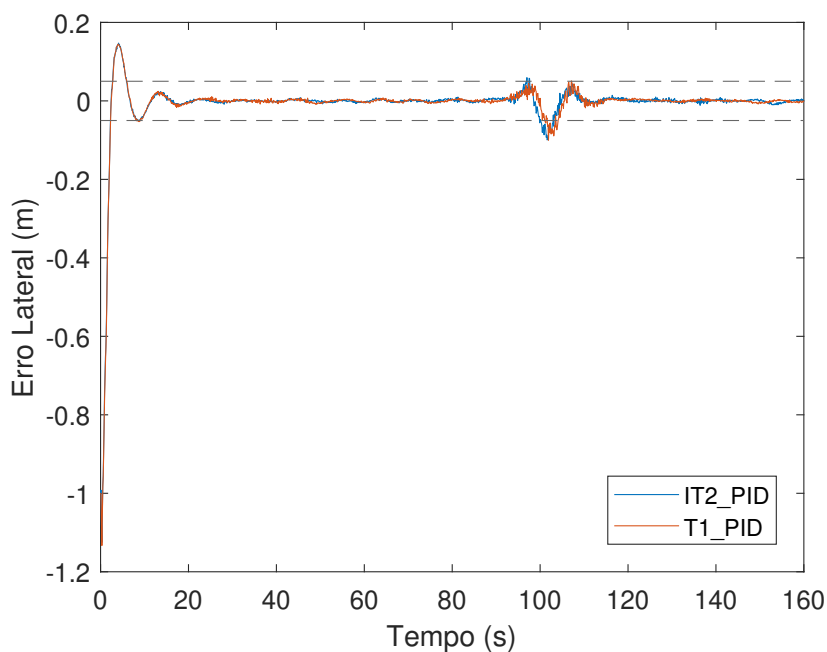
4.1.4 Controle IT2_PID em missão de seguimento das linhas de distribuição - Cenário simulado.

O último teste realizado em ambiente simulado consistiu em uma missão de seguimento de linhas. Para isso, utilizou-se o circuito completo ilustrado na Figura 27, composto por três torres interligadas por dois conjuntos de três cabos de distribuição de energia. O drone inicia a missão posicionado a 1 metro do cabo central, com um desvio angular de 0.75 rad e velocidade de avanço de 0.4 m/s. A missão tem duração total de 160 segundos, sendo que, por volta dos 90 segundos, ocorre uma inflexão na trajetória, causada pela disposição das torres. Nesta etapa,

foram realizadas 50 execuções para cada uma das duas abordagens *fuzzy*, a fim de compará-las.

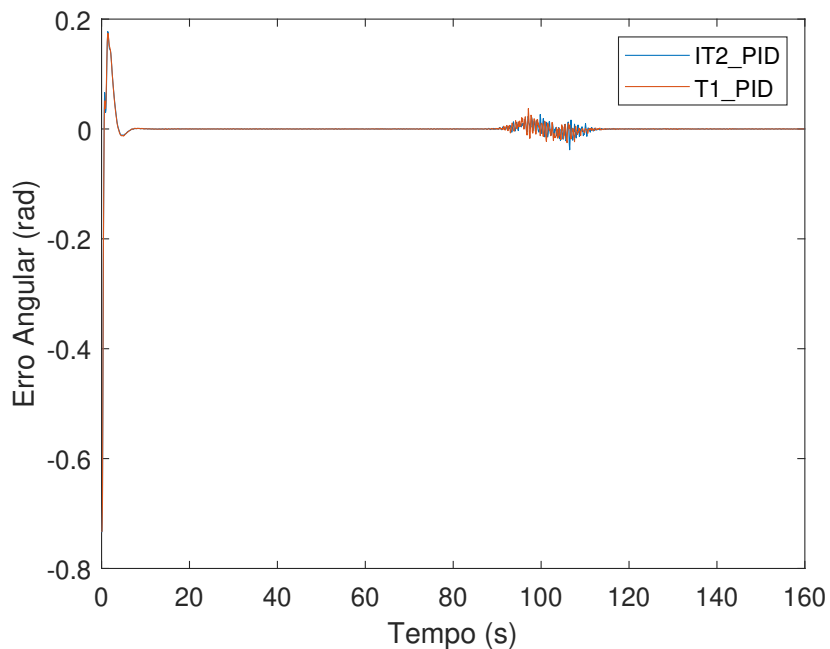
Os valores médios para os erros lateral e angular durante a missão podem ser vistos nas Figuras 33 e 34. É possível observar que o drone conseguiu seguir a trajetória de maneira satisfatória, se estabilizando após o início da missão e mantendo o erro em níveis baixos durante a porção retilínea. O ponto em que as linhas mudam de orientação pode ser visto pelas oscilações nas curvas de erro. É importante salientar que, em locais onde ocorre a interseção entre os cabos, existe um certo grau de imprecisão na estimação dos parâmetros da trajetória, fazendo com que este seja um ponto interessante para averiguar a melhor capacidade da lógica *fuzzy* do tipo 2 em lidar com esse tipo de incerteza se comparado ao tipo 1. Após a curva, o drone conseguiu se alinhar novamente e seguiu a missão até o fim de maneira satisfatória.

Figura 33 – Comportamento do erro lateral durante missão de seguimento das linhas de distribuição em cenário simulado.



Fonte: Elaborado pelo autor.

Figura 34 – Comportamento do erro angular durante missão de seguimento das linhas de distribuição em cenário simulado.



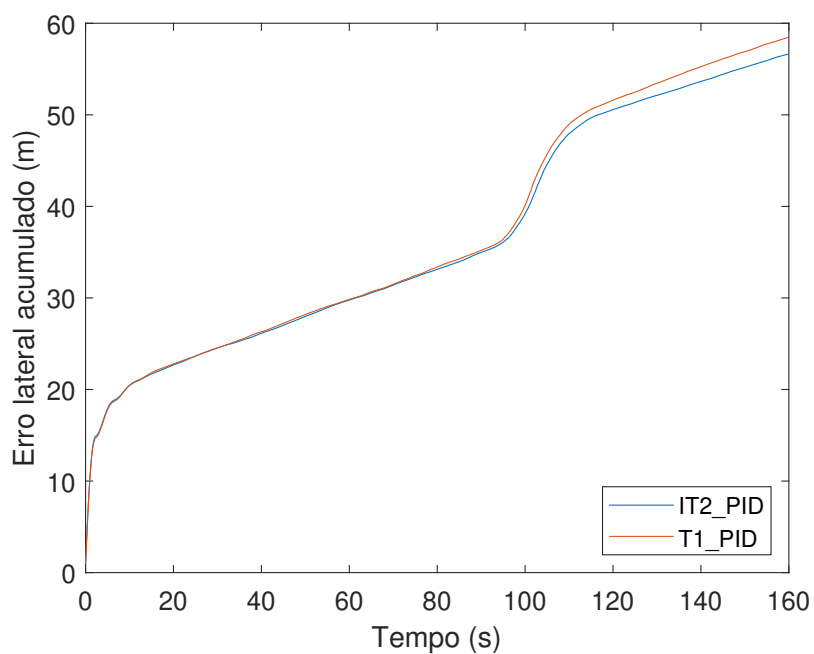
Fonte: Elaborado pelo autor.

Os resultados numéricos das simulações da missão de seguimento podem ser vistos na Tabela 8. É possível observar que o IT2_PID se sobressai em relação ao T1_PID para os dois erros, tanto em valores médios como no desvio padrão das execuções indicando, novamente, a vantagem da utilização da lógica *fuzzy* do tipo 2 em aplicações nas quais a inferência dos dados do sistema não é completamente precisa. Para verificar a evolução dos erros durante a missão, é importante analisar os gráficos de erro acumulado, que podem ser vistos nas Figuras 35 e 36. As curvas se mantêm coincidentes durante a porção retilínea da trajetória. Ao chegar no ponto de inflexão, os erros utilizando o controlador T1_PID aumentam em maior grau se comparado ao IT2_PID devido ao maior grau de incerteza na estimação.

Tabela 8 – Resultados quantitativos para a missão de seguimento das linhas em cenário simulado

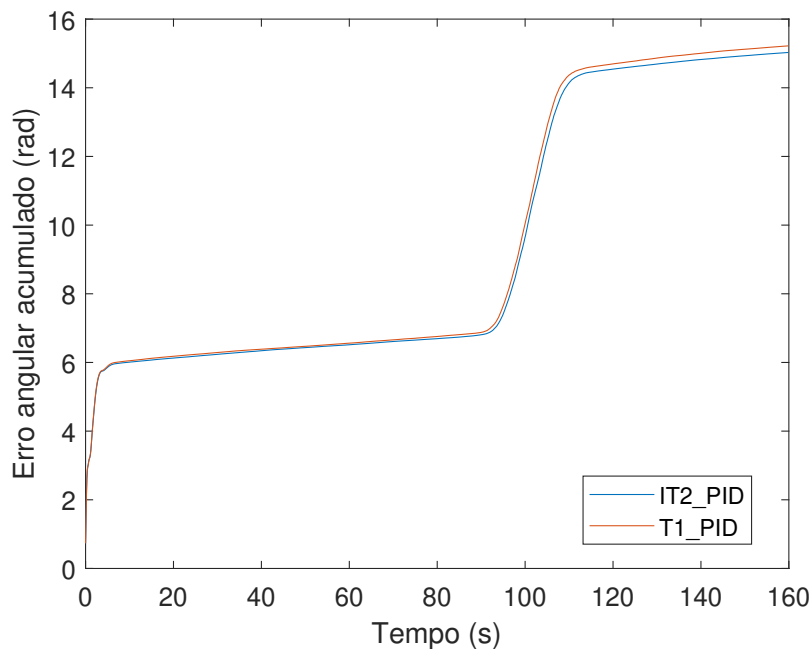
Controlador	IAE_Z	IAE_φ _r
IT2_PID	5.6645 ± 0.3721	1.5028 ± 0.1112
T1_PID	5.8454 ± 0.4960	1.5221 ± 0.1557

Figura 35 – Comportamento do erro lateral acumulado durante missão de seguimento das linhas de distribuição em cenário simulado.



Fonte: Elaborado pelo autor.

Figura 36 – Comportamento do erro angular acumulado durante missão de seguimento das linhas de distribuição em cenário simulado.



Fonte: Elaborado pelo autor.

4.2 Resultados em ambiente real

Esta seção traz os testes envolvendo toda a metodologia proposta, tanto para a detecção das linhas de distribuição em cenários reais quanto a utilização do controlador IT2_PID para as missões de alinhamento, seguimento e inspeção. Inicialmente, foi realizado um comparativo entre os modelos baseados na rede YOLOv8, com seu antecessor YOLOv5 para a tarefa de segmentação de objetos, mostrando a viabilidade dessa técnica considerando os requisitos temporais do projeto. Após a escolha do modelo de rede adequado, ele foi empregado em conjunto com os algoritmos de estimativa da trajetória e a malha de controle para demonstrar o funcionamento de todo o processo por meio de um drone físico sobrevoando cabos de distribuição reais. Além disso, durante o processo de seguimento, elementos como postes, cruzetas e isoladores são identificados e apresentados ao operador, que pode pausar o seguimento autônomo para inspecionar esses elementos de forma

minuciosa.

4.2.1 Análise do modelo YOLOv8 para identificação das linhas de distribuição.

Para a determinação do modelo YOLO implementado na metodologia final para inspeção em cenários reais, é necessário verificar as métricas de desempenho das redes após o treinamento das mesmas, visando encontrar um modelo que detecte as linhas com confiabilidade, ao mesmo tempo que não exija grandes recursos computacionais da máquina que será utilizada para os testes. Após o treinamento dos modelos, é realizada a etapa de validação, na qual um conjunto de imagens não utilizadas no processo de treinamento é inserido na rede com o intuito de verificar a capacidade de generalização da detecção em cenários não observados no processo de aprendizado. Os 5 modelos de YOLOv8 para a tarefa de segmentação foram comparados com os modelos análogos utilizando a rede YOLOv5, com o objetivo de verificar se existem melhorias que justifiquem o uso da técnica mais recente, de acordo com os parâmetros de detectabilidade e custo computacional. As métricas de comparação adotadas são: Revocação (R), Precisão (P), Precisão Média (AP) e Média da Precisão Geral (mAP), que são calculadas de acordo com as Equações 4.1,4.2,4.3:

$$P = \frac{TP}{TP + FP} \quad (4.1)$$

$$R = \frac{TP}{TP + FN} \quad (4.2)$$

$$AP = \sum_{k=1}^{k=n} [R(k) - R(k+1)] * P(k) \quad (4.3)$$

$$mAP = \frac{1}{N} \sum_{k=1}^{k=N} AP_k$$

Em que TP são os Verdadeiros Positivos, FP são os Falsos Positivos, FN são os Falsos Negativos, n é o número de limites, N é o número de classes, AP é a precisão média e mAP é a média da precisão média.

Além disso, duas variações amplamente utilizadas dessa métrica são o mAP@50 e o mAP@50:95. O mAP@50 corresponde ao valor médio de AP calculado considerando um limiar de interseção sobre união (IoU) igual a 0.5, isto é, uma detecção é considerada correta quando a sobreposição entre a *bounding box* predita e a real é superior a 50%. Já o mAP@50:95 representa a média do AP calculada para múltiplos limiares de IoU variando de 0.5 a 0.95, com incrementos de 0.05, fornecendo assim uma avaliação mais rigorosa do desempenho do modelo.

Os resultados podem ser vistos na Tabela 9, tanto para as métricas relacionadas à caixa delimitadora (Box) quanto para a máscara de segmentação do objeto (Mask).

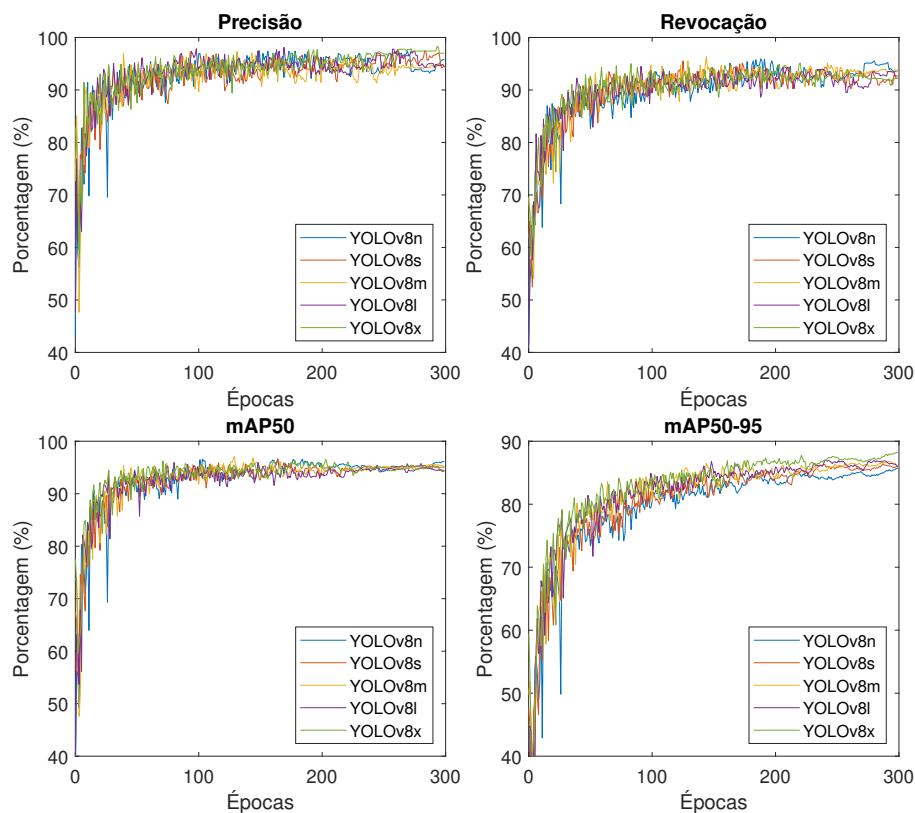
Tabela 9 – Métricas de Performance para modelos YOLO-seg para tarefa de segmentação das linhas de distribuição

	Box				Mask				Tempo
	P	R	mAP@50	mAP@50: 95	P	R	mAP@50	mAP@50: 95	
YOLOv5n-seg	0.960	0.870	0.942	0.853	0.931	0.841	0.913	0.779	6.1ms
YOLOv8n-seg	0.967	0.947	0.974	0.913	0.954	0.936	0.961	0.855	6.3ms
YOLOv5s-seg	0.945	0.908	0.953	0.869	0.937	0.848	0.916	0.798	11.0ms
YOLOv8s-seg	0.976	0.939	0.980	0.917	0.962	0.915	0.950	0.863	13.6ms
YOLOv5m-seg	0.965	0.895	0.961	0.895	0.951	0.866	0.930	0.815	20.3ms
YOLOv8m-seg	0.959	0.959	0.982	0.935	0.947	0.937	0.955	0.868	24.8ms
YOLOv5l-seg	0.961	0.909	0.957	0.904	0.944	0.893	0.942	0.832	32.7ms
YOLOv8l-seg	0.983	0.943	0.977	0.939	0.970	0.930	0.951	0.865	40.9ms
YOLOv5x-seg	0.971	0.913	0.964	0.911	0.953	0.901	0.949	0.851	65.3ms
YOLOv8x-seg	0.987	0.943	0.977	0.945	0.965	0.923	0.950	0.880	68.8ms

A comparação das métricas de performance indica que o YOLOv8x teve um desempenho superior nos resultados do mAP50:95 tanto para a detecção do objeto quanto para a segmentação da área compreendida pelas linhas. O YOLOv8m-seg apresentou os melhores resultados para as métricas de Revocação e mAP@50, enquanto a Precisão se sobressaiu nos modelos Large (l) e extra-large (x). Uma comparação direta entre modelos análogos mostra que a arquitetura YOLOv8 apresentou melhorias em todas as métricas de detecção e segmentação de objetos. Os valores de mAP50:95 indicam que a versão mais pesada do YOLOv5-seg(x) produz resultados comparáveis aos da versão mais leve do YOLOv8-seg(n), apesar do tempo de inferência ser dez vezes maior. Desta forma, as análises subsequentes se concentrarão exclusivamente na arquitetura YOLOv8.

A evolução das métricas de desempenho durante o treinamento da rede

Figura 37 – Comparação das métricas de performance para o treinamento da rede YOLOv8-seg



Fonte: Elaborado pelo autor.

pode ser vista na Figura 37. O processo de aprendizado da rede começa a convergir a partir da 50^a época para as métricas de precisão, revocação e mAP@50, enquanto o mAP@50:95 converge a partir da 150^a época, mantendo-se estável até o fim do processo. Essa informação indica que a escolha dos hiperparâmetros (foram utilizados os valores padrão do yolov8) e a quantidade de épocas utilizadas para o treinamento são adequadas para o banco de imagens que foi construído.

O último passo para a escolha do modelo a ser utilizado na metodologia proposta é verificar o tempo necessário para a inferência de uma imagem, considerando a capacidade computacional da máquina portátil a ser utilizada nos testes em campo, que possui um processador Intel Core i5 13420H com 8 GB de

memória RAM e uma placa gráfica NVIDIA GeForce RTX 3050. Assim, o processo de inferência foi executado novamente, desta vez na máquina a ser utilizada nos testes de controle. O tempo necessário para a segmentação do objeto em cada modelo pode ser visto na Tabela 10. Considerando que a malha de controle opera a uma frequência de 10 Hz (100 ms por ciclo), apenas uma fração desse tempo está disponível para a etapa de inferência do modelo, uma vez que o sistema também precisa executar o cálculo dos parâmetros da trajetória e dos sinais de controle. Dessa forma, o tempo de processamento torna-se um fator crítico para a escolha do modelo. Além disso, conforme apresentado na Tabela 10, a utilização de modelos mais complexos, como o YOLOv8m-seg, não resulta em melhorias significativas de desempenho quando comparada ao modelo mais leve YOLOv8n-seg. Portanto, considerando o compromisso entre custo computacional e desempenho, a escolha mais adequada para o sistema proposto é o modelo YOLOv8n-seg.

Tabela 10 – Tempo necessário para inferência da YOLOv8-seg na máquina utilizada nos testes de campo.

Modelo	Tempo computacional
YOLOv8n-seg	5.0 ms
YOLOv8s-seg	10.4 ms
YOLOv8m-seg	23.2 ms
YOLOv8l-seg	38.3 ms
YOLOv8x-seg	58.7 ms

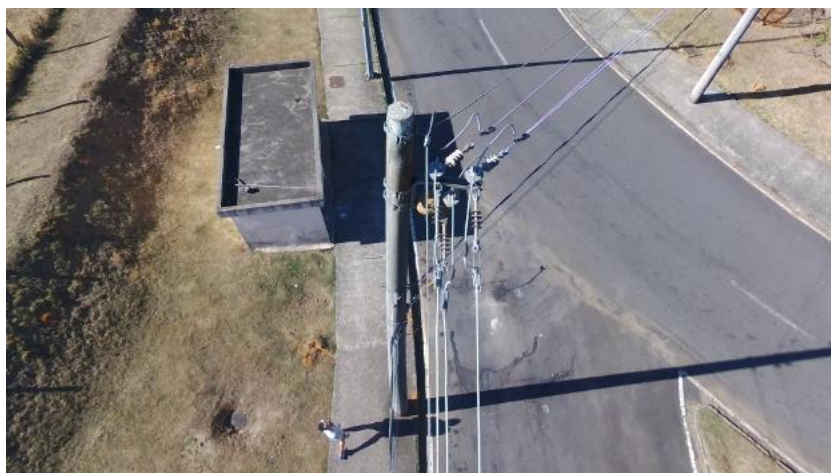
4.2.2 Análise do modelo YOLOv8 para a detecção dos elementos da rede de distribuição

Além da identificação e segmentação das linhas, é importante avaliar se o modelo de detecção dos elementos das redes de distribuição é capaz de prover feedback visual ao operador em tempo quase real. Contudo, observa-se que os objetos de interesse apresentam variações significativas de escala e orientação devido à dinâmica de voo, fator que é agravado pela degradação progressiva da qualidade da imagem recebida em solo. Como ilustrado na Figura 38, há uma disparidade nítida entre o dado original e o transmitido. Enquanto a imagem armazenada na memória interna preserva os detalhes finos da infraestrutura (Figura 38b), o

sistema Bebop Autonomy introduz artefatos severos de compressão e redução de resolução para viabilizar o envio via Wi-Fi em baixa latência (Figura 38a). Essa perda de fidelidade, intensificada pelo aumento da distância entre o operador e o drone, impacta diretamente a confiabilidade das detecções e a percepção situacional durante a inspeção.



(a) Streaming via Bebop Autonomy: Note a baixa resolução e artefatos de compressão devido à transmissão via Wi-Fi.



(b) Arquivo Original: Imagem processada e salva diretamente na memória interna com fidelidade total.

Figura 38 – Comparativo de perda de qualidade de imagem do Parrot Bebop 2. Acima, o sinal recebido pelo computador via rede; abaixo, a captura direta da memória do drone.

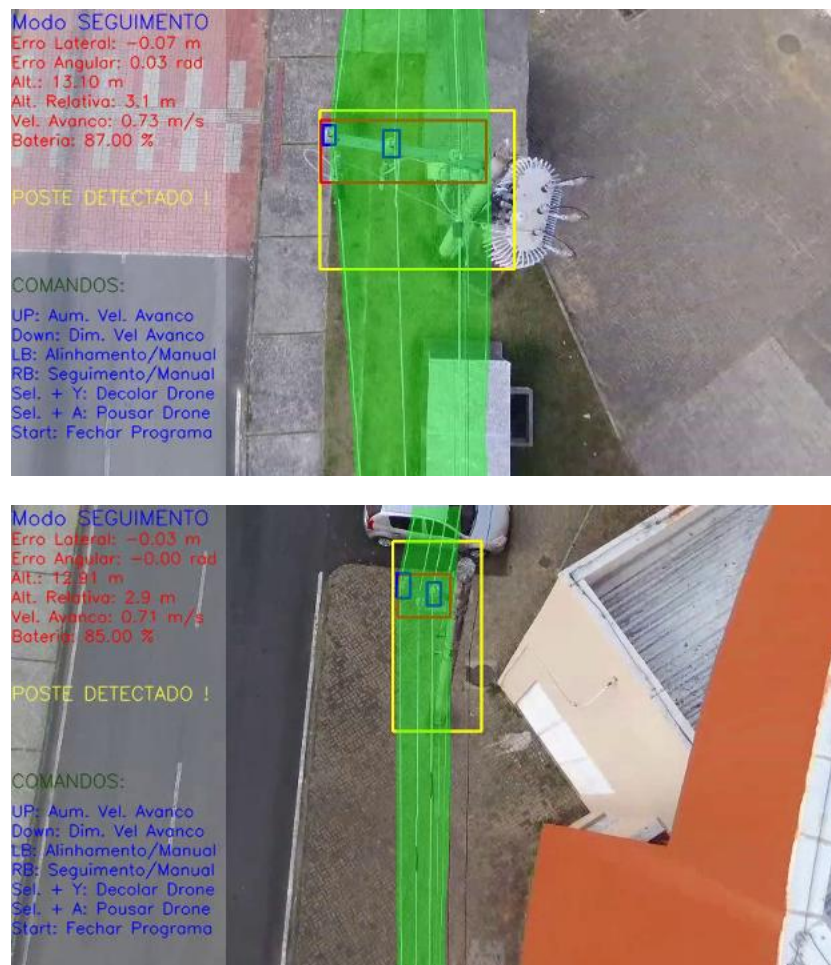
Diante dessas restrições computacionais e de comunicação, optou-se pelo uso do modelo YOLOv8 na sua variante nano (YOLOv8n), visando manter baixo custo computacional e garantir execução em paralelo com os módulos de controle e segmentação das linhas. Além disso, o sistema deve ser capaz de enviar comandos de retorno ao drone a uma frequência de 10 Hz, o que impõe requisitos adicionais de eficiência. A Tabela 11 apresenta os resultados de treinamento do modelo, tanto de forma agregada quanto por classe, considerando os diferentes tipos de objetos presentes nas linhas de distribuição.

Tabela 11 – Métricas de Performance para modelo YOLOv8 para detecção de elementos do sistema de distribuição

Classe	P	R	mAP50	mAP50:95
Total	0.902	0.847	0.886	0.540
Cruzeta	0.915	0.898	0.915	0.578
Isolador	0.857	0.706	0.774	0.320
Poste	0.934	0.938	0.970	0.722

É possível notar que o modelo apresentou métricas satisfatórias, com uma precisão de 0.9 e revocação de 0.85. Como os isoladores são os menores objetos a serem detectados, era de se esperar que suas métricas fossem as piores do modelo proposto. Outro ponto a se considerar é que, como o modelo foi treinado e as inferências foram realizadas com a resolução da imagem de 640x640, de forma a manter o esforço computacional reduzido, a rede apresenta uma dificuldade natural em identificar padrões pequenos na imagem. Apesar disso, os postes apresentaram uma boa detectabilidade e, desta forma, foram adicionados alertas visuais desse elemento para as tarefas de inspeção das linhas, conforme mostrado na Figura 39. Nota-se que nem todos os isoladores (retângulos azuis) foram detectados corretamente, o que varia de frame a frame, conforme o seguimento ocorre.

Figura 39 – Detecção dos elementos na linha de distribuição na missão de seguimento



Fonte: Elaborado pelo autor.

4.2.3 Testes de controle em ambiente real

Feitas as análises que comprovam a eficácia da técnica utilizada para a detecção das linhas de distribuição e dos outros elementos da rede em um tempo adequado, o próximo passo é integrar os sistemas de identificação das linhas com a malha de controle para que o drone consiga executar a missão de seguimento dos alvos visuais. Para esse propósito, foi escolhida uma localidade na Faculdade de Engenharia da Universidade Federal de Juiz de Fora com pouco trânsito de pessoas e veículos para que os testes pudessem ser realizados sem riscos adicionais.

O drone é posicionado a uma altitude de 13.5 metros com relação ao solo (altitude relativa de 3.5 metros dos cabos) manualmente com o auxílio de um *joystick* conectado à máquina que gerencia a missão. Para cada teste de alinhamento, o drone é posicionado de forma que exista um erro a ser corrigido pelos controladores autônomos, considerando o objetivo de manter as linhas próximas ao centro da imagem e perpendiculares à porção longitudinal da mesma, conforme mostrado na Figura 40. Para cada missão, são feitas 10 execuções, com 30 segundos de duração cada. O sinal para o início de cada execução é dado pelo operador, que tem a capacidade de abortar a missão, voltando ao controle manual a qualquer momento.

4.2.3.1 Testes de alinhamento lateral

Neste experimento, o drone é posicionado a 1.3 m do ponto médio entre as linhas e o objetivo do controlador é mantê-lo exatamente acima das mesmas. O comportamento do erro médio durante a missão pode ser visto na Figura 41, e as métricas de desempenho de controle são mostradas na Tabela 12. O drone conseguiu cumprir o objetivo, corrigindo o erro rapidamente e mantendo-se próximo à posição de referência até o fim da missão, com um tempo de subida médio de 3.28 segundos e entrando na faixa de regime permanente em aproximadamente 8.6 segundos. O vídeo de uma das execuções do alinhamento lateral pode ser visto através do *link*. Em aproximadamente 0:35, o drone inicia o alinhamento, mantendo as linhas próximas ao centro da imagem até o fim do teste.

Tabela 12 – Métricas de performance do controlador IT2_PID para alinhamento lateral.

Controlador lateral	
Pico máximo	$20.67 \pm 3.65\%$
Tempo de subida	$3.28 \pm 0.38s$
Tempo de acomodação	$4.92 \pm 0.82s$

O processo de adaptação dos ganhos do controlador IT2_PID para esta missão pode ser visto na Figura 42. É possível notar que o ganho K_p sofre uma leve redução durante os períodos em que o erro se aproxima de 0 e mantém-se relativamente constante durante o período permanente do sistema. O ganho K_d

Figura 40 – Ponto de partida para os testes de alinhamento em um ambiente real.

(a) Alinhamento lateral

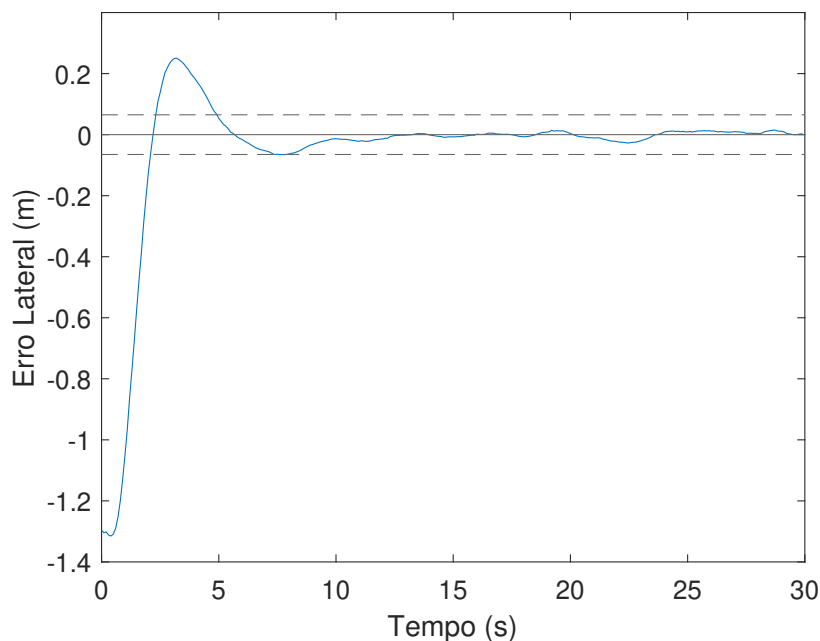


(b) Alinhamento angular



Fonte: Elaborado pelo autor.

Figura 41 – Evolução temporal do erro para a missão de alinhamento lateral.



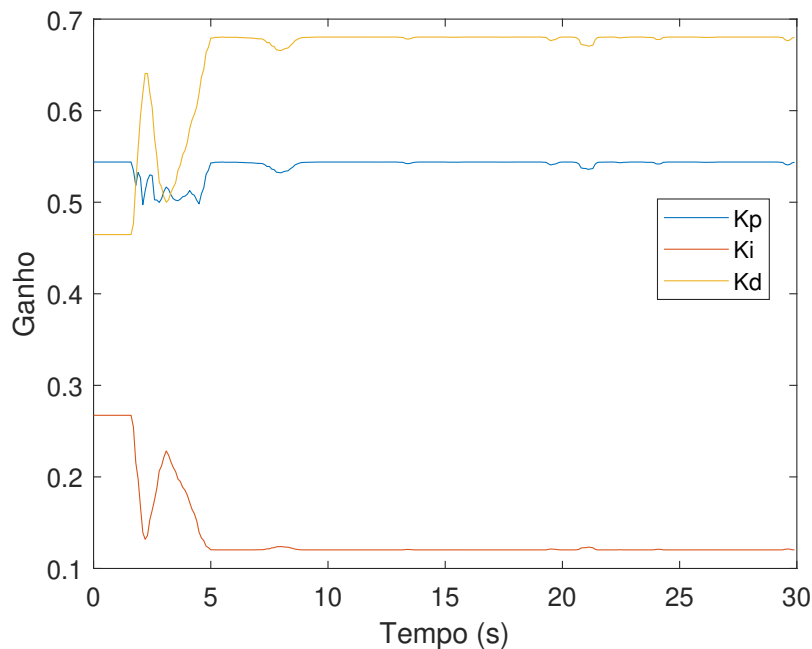
Fonte: Elaborado pelo autor.

começa com uma baixa magnitude, permitindo que o erro atinja o referencial rapidamente e oscila durante o período transitório, servindo como um freio para a parcela proporcional, atingindo valores mais altos durante o período permanente, de forma que pequenas variações sejam rapidamente tratadas. O ganho K_i inicia com grande magnitude, uma vez que o erro é grande e precisa de uma ação de controle forte para atingir o referencial. Ao chegar próximo do erro nulo, a ação da parcela integral pode ser reduzida, sendo que seu aumento é necessário somente caso o erro apresente um comportamento anômalo de acréscimo devido a fatores externos.

4.2.3.2 Testes de alinhamento angular

Para os testes de alinhamento angular, o drone é posicionado acima das linhas com um desvio de orientação de 0.6 rad das mesmas. O objetivo da missão é deixar o drone alinhado paralelamente aos cabos. O comportamento do erro e

Figura 42 – Evolução temporal dos ganhos do controlador IT2_PID para a missão de alinhamento lateral.



Fonte: Elaborado pelo autor.

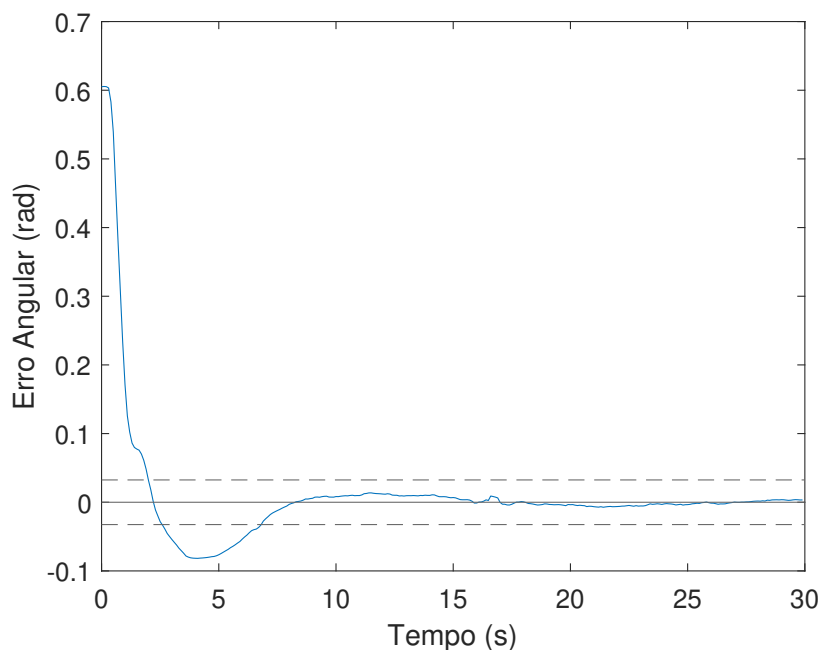
as métricas de performance podem ser vistos na Figura 43 e na Tabela 13. Assim como para a missão de alinhamento lateral, o erro foi reduzido rapidamente e, em 6.8 s, atingiu a faixa de estado terminal, mantendo-se nela até o fim da missão. Um vídeo mostrando uma das execuções da missão de alinhamento angular pode ser acessado pelo *link*.

Tabela 13 – Métricas de performance do controlador IT2_PID para alinhamento angular.

Controlador angular	
Pico máximo	13.84 ± 0.51 %
Tempo de subida	4.26 ± 0.20 s
Tempo de acomodação	6.82 ± 1.24 s

A variação dos ganhos durante o teste pode ser vista na Figura 44. A

Figura 43 – Evolução temporal do erro para a missão de alinhamento angular



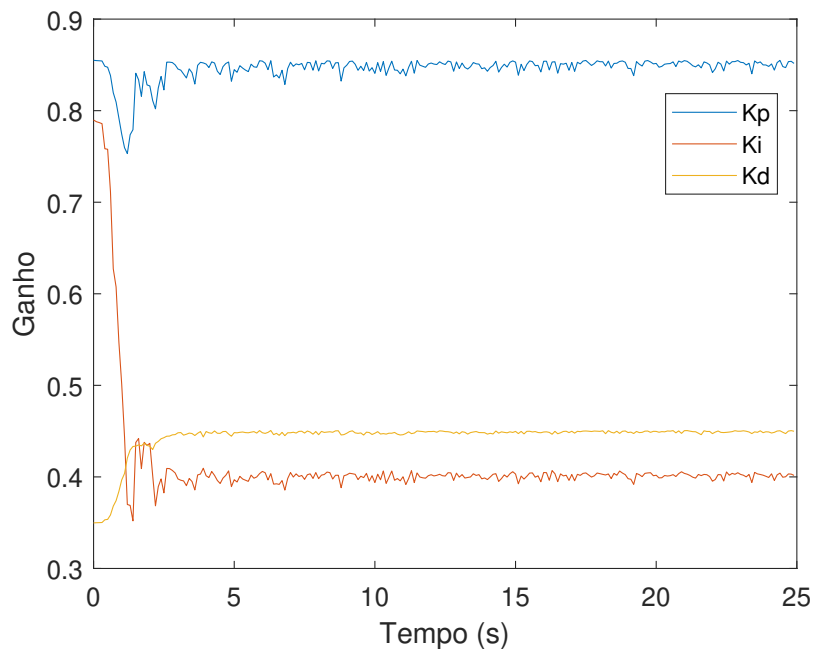
Fonte: Elaborado pelo autor.

magnitude de K_p para este controle é superior se comparada aos outros dois ganhos. O comportamento temporal dos mesmos segue as características observadas para o alinhamento lateral, o que pode ser explicado pela semelhança do conjunto de regras utilizado para os dois casos, fazendo com que o sistema se comporte da mesma maneira em cenários nos quais os erros de estado inicial são elevados.

4.2.3.3 Testes de seguimento das linhas de distribuição

Após comprovar a eficácia da metodologia proposta em alinhar o drone tanto lateralmente quanto em relação à orientação das linhas, o próximo passo é verificar se a malha de controle é capaz de manter o drone posicionado acima das linhas conforme o mesmo avança com uma velocidade linear sobre elas, em um ambiente com passagem entre postes de distribuição, mudanças de orientação e disposição dos cabos, além de variação nos fundos das imagens com passagem de veículos e pedestres. Para este propósito, os testes de seguimento foram realizados

Figura 44 – Evolução temporal dos ganhos do controlador IT2_PID para a missão de alinhamento angular.



Fonte: Elaborado pelo autor.

em dois cenários, com o início e fim das missões controlados pelo operador por meio de um *joystick*. Os resultados são apresentados a seguir.

4.2.3.3.1 Cenário 1

O primeiro cenário escolhido apresenta um percurso majoritariamente linear, contendo quatro postes, conforme mostrado na Figura 45. O círculo azul indica a posição inicial do teste, os círculos amarelos indicam os postes intermediários que serão seguidos, e o ponto verde indica o último poste visitado, correspondendo ao fim da missão.

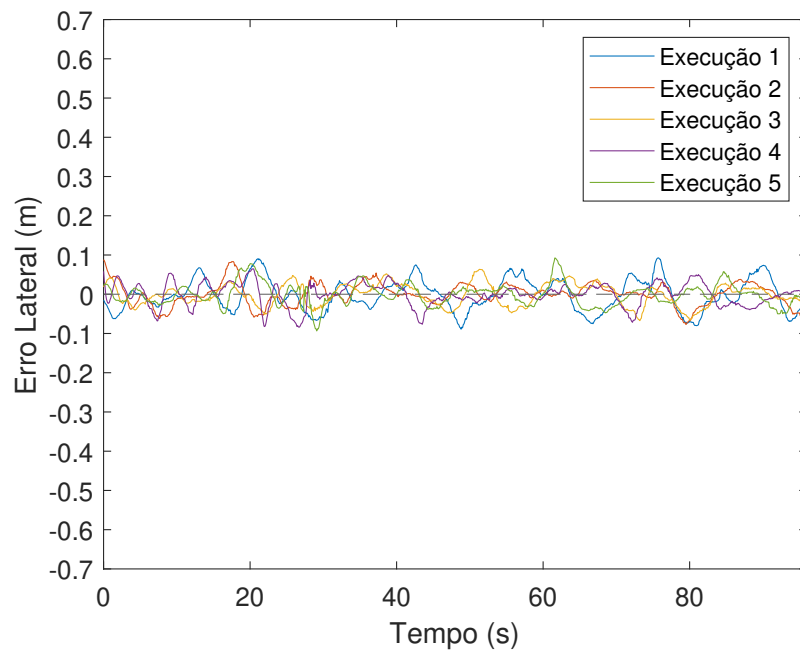
Figura 45 – Trajetória para o Teste de Seguimento - Cenário 1



Fonte: Elaborado pelo autor.

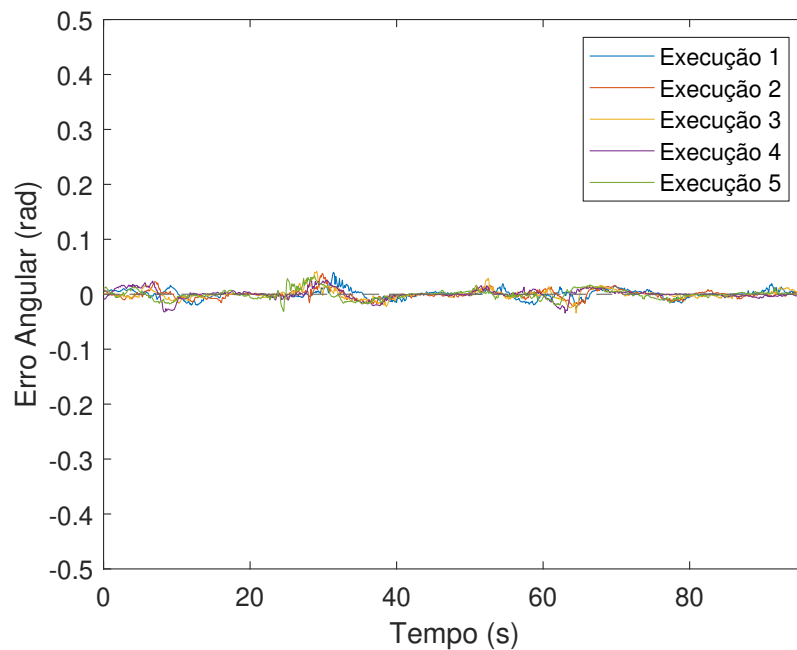
Pela característica do trajeto, é esperado que os erros mantenham-se próximos de zero, com o drone sendo capaz de corrigir a posição e a orientação, caso necessário. Os testes foram executados 5 vezes, e o perfil dos erros pode ser visto nas Figuras 46 e 47. É possível notar que a magnitude dos erros se mantém próxima do referencial durante todo o percurso, com o erro lateral não ultrapassando a marca de 10 cm e o erro angular com valores máximos de 0.06 rad para todas as execuções. É possível observar pequenas variações no erro angular por volta de 30 e 60 segundos que indicam pequenas mudanças na orientação da trajetória devido à presença dos postes intermediários (em amarelo na Figura 45). Além disso, existem pequenas oscilações no erro lateral. No entanto, o drone não perdeu o referencial e conseguiu completar as 5 execuções de maneira satisfatória.

Figura 46 – Erro lateral para teste de seguimento - Cenário 1



Fonte: Elaborado pelo autor.

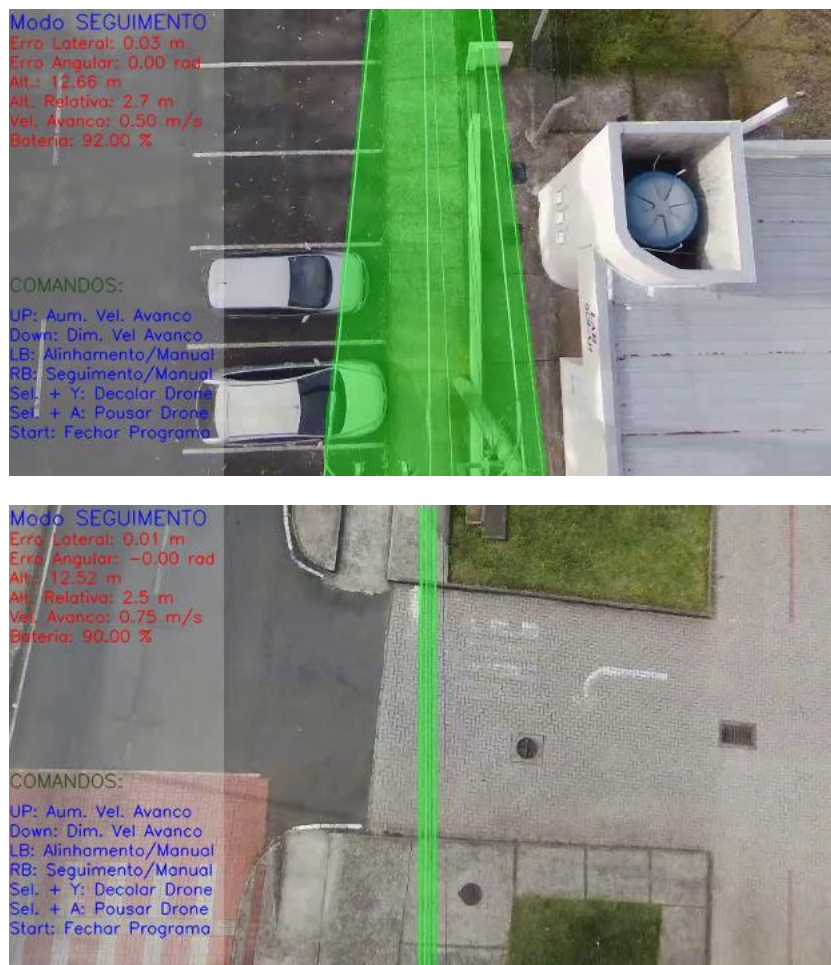
Figura 47 – Erro angular para teste de seguimento - Cenário 1



Fonte: Elaborado pelo autor.

Imagens da missão de seguimento no cenário 1 podem ser vistas na Figura 48 e através do *link*. É possível observar que o modelo de segmentação é capaz de detectar as linhas de distribuição em diferentes ambientes e disposições dos cabos de energia. O operador dispõe das informações importantes durante a missão, podendo terminá-la pressionando um botão do controle conectado ao computador que processa a operação.

Figura 48 – Teste de seguimento em ambiente real - Cenário 1



Fonte: Elaborado pelo autor.

4.2.3.3.2 Cenário 2

O segundo cenário escolhido para os testes é apresentado na Figura 49. É possível observar que, entre o primeiro e o segundo postes intermediários, existe uma mudança abrupta na orientação da trajetória, o que apresenta um desafio adicional à malha de controle do drone.

Novamente, 5 execuções foram realizadas, e os perfis dos erros lateral e angular durante os testes podem ser vistos nas Figuras 50 e 51. É possível observar

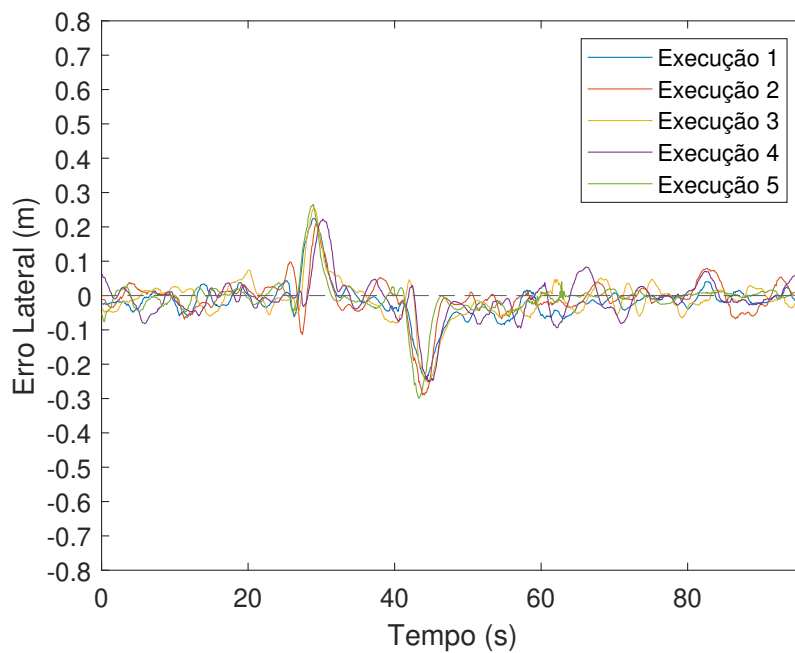
Figura 49 – Trajetória para o Teste de Seguimento - Cenário 2



Fonte: Elaborado pelo autor.

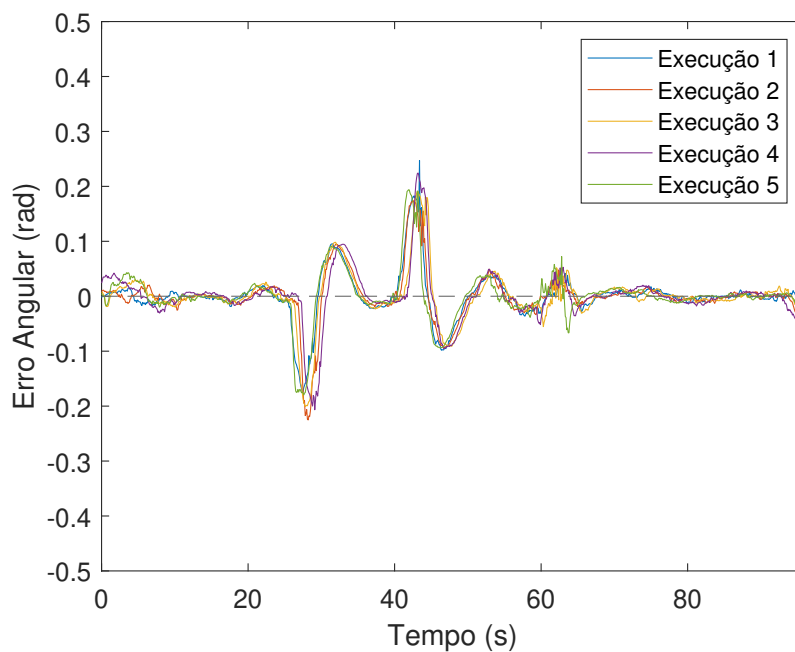
que o drone foi capaz de seguir a trajetória de maneira adequada durante as execuções. Nos instantes 26s e 43s, o drone passou próximo aos dois primeiros postes intermediários e foi capaz de corrigir tanto a posição lateral quanto a orientação em relação às linhas de distribuição, mantendo-se próximo a elas até o fim do trajeto.

Figura 50 – Erro lateral para teste de seguimento - Cenário 2



Fonte: Elaborado pelo autor.

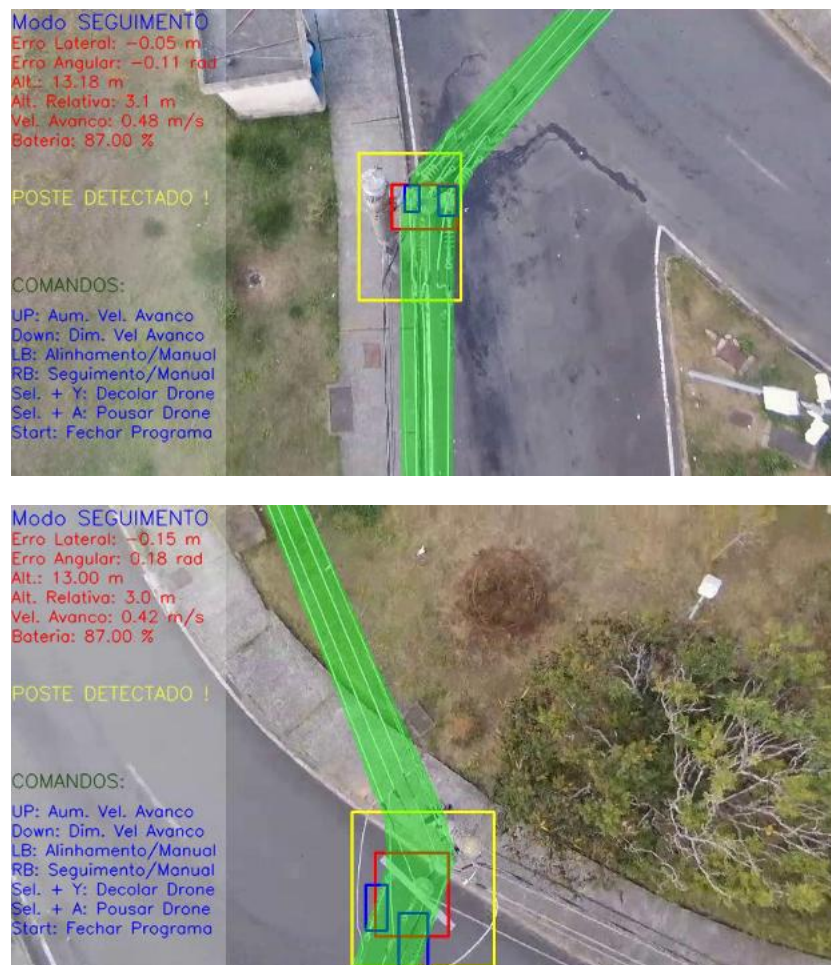
Figura 51 – Erro angular para teste de seguimento - Cenário 2



Fonte: Elaborado pelo autor.

Exemplos visuais do teste de seguimento no segundo cenário podem ser vistos por meio do *link*, enquanto a passagem do drone pelos postes intermediários pode ser vista na Figura 52. Além das linhas (região em verde), outros elementos também foram detectados, gerando um alerta de 'POSTE DETECTADO !', abaixo dos dados de telemetria da missão.

Figura 52 – Teste de seguimento em ambiente real - Cenário 2



Fonte: Elaborado pelo autor.

4.2.3.4 Testes de inspeção das linhas de distribuição

Como teste final da metodologia proposta, foi realizada uma missão de inspeção das linhas de distribuição de energia. Para tal, o drone segue pelo cenário 1 no modo autônomo. Ao chegar próximo do segundo poste intermediário, representado pelo círculo vermelho na Figura 53, o operador ativa o modo de inspeção manual para obter imagens diversificadas dos elementos a serem analisados. Ao término da inspeção, o operador deve retornar ao modo de seguimento autônomo e, para isso, o controlador deve ser capaz de realinhar o drone com os cabos de

energia de forma que o drone termine a missão no poste representado pelo círculo verde.

Figura 53 – Trajetória para o Teste de Inspeção



Fonte: Elaborado pelo autor.

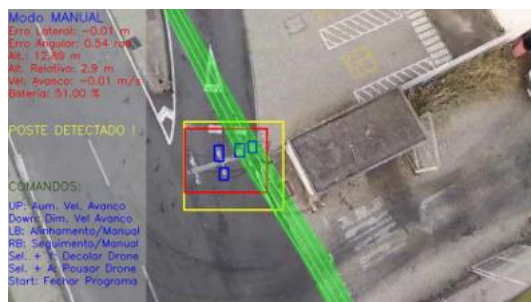
Um vídeo da missão de inspeção pode ser acessado por meio do *link*. Exemplos visuais do início, da inspeção manual e do fim da missão são apresentados na Figura 54. No canto superior esquerdo, é apresentado o modo atual da missão e, mais abaixo, estão os comandos que o operador pode usar durante a operação. A segmentação das linhas é útil para alertar o operador caso as mesmas saiam do campo de visão do drone durante a operação manual.

Figura 54 – Exemplos visuais da missão de inspeção das linhas de distribuição

(a) Início da missão



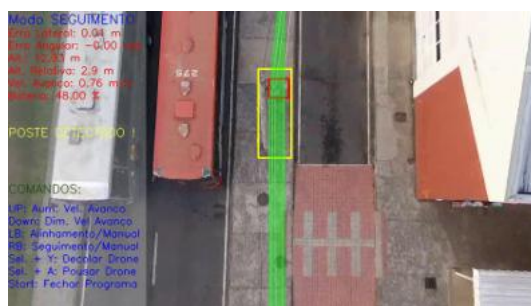
(b) Inspeção manual - Exemplo 1



(c) Inspeção manual - Exemplo 2



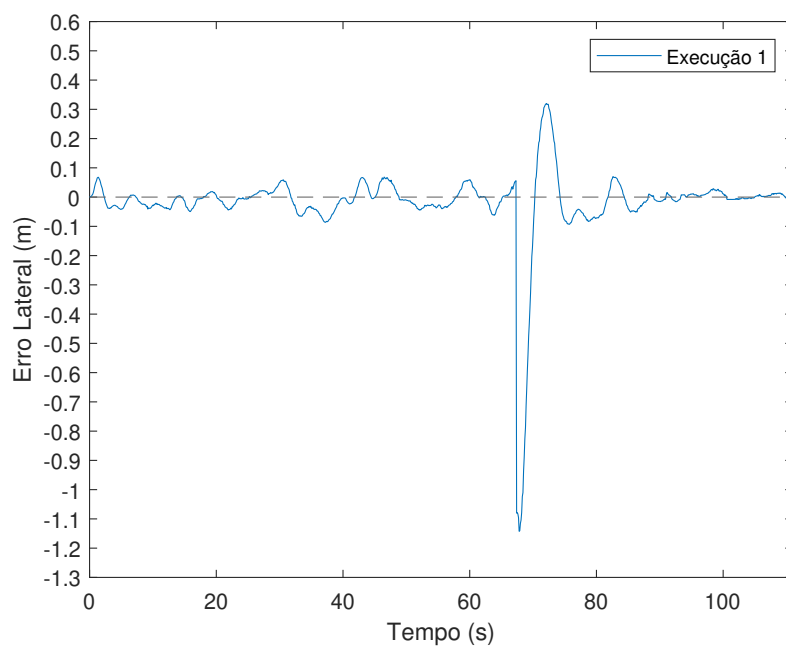
(d) Fim da missão



Fonte: Elaborado pelo autor.

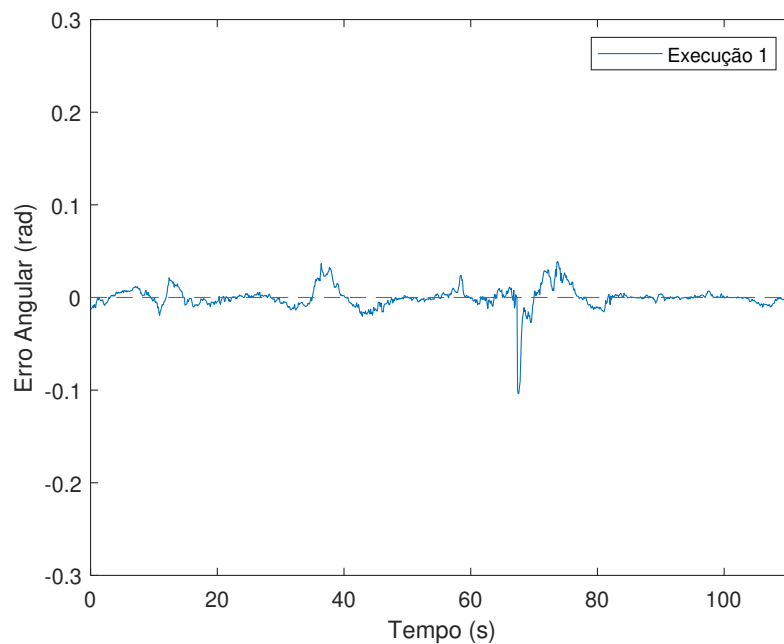
Os perfis dos erros lateral e angular durante a missão de inspeção podem ser vistos nas Figuras 55 e 56. É possível notar que, no instante 68s, existe uma descontinuidade, uma vez que o gráfico apresenta somente as leituras durante o modo autônomo, no qual os controladores IT2_PID atuam no sistema. Ao término da inspeção manual, o drone situava-se a cerca de 1.15m das linhas, com um erro de orientação de 0.1rad e, com a volta do modo autônomo, os erros foram rapidamente corrigidos conforme o drone seguia a trajetória.

Figura 55 – Erro lateral para teste de inspeção



Fonte: Elaborado pelo autor.

Figura 56 – Erro angular para teste de inspeção



Fonte: Elaborado pelo autor.

4.2.4 Discussões

Este capítulo teve como objetivo apresentar os resultados que corroboram a metodologia proposta para o alinhamento e seguimento de linhas de distribuição por meio de drones. O controle IT2_PID foi capaz de realizar as missões tanto em um cenário criado no simulador Gazebo quanto em uma linha de distribuição real, por meio do drone Bebop Parrot 2 em missões de alinhamento, seguimento e inspeção.

O objetivo inicial dos testes simulados foi trazer uma análise inicial dos algoritmos construídos, sem o risco de acidentes e a necessidade de lidar com problemas como falhas no sistema de imagem e recarga de baterias. O foco dos testes iniciais em um cenário controlado é comprovar a eficácia de controladores híbridos IT2_PID em uma aplicação envolvendo seguimento de caminhos visuais. O controlador proposto foi comparado com sua variante usando lógica *fuzzy* do tipo

1 e controladores PID clássicos de ganhos fixos. Os resultados para o alinhamento angular e lateral mostram que abordagens adaptativas conseguiram levar o drone à posição desejada com menores oscilações no período transitório do movimento, devido à capacidade de alterar o comportamento dos ganhos de acordo com o estado atual da planta. Os resultados para a missão de inspeção mostraram que a variante IT2_PID se sobressaiu em relação ao T1_PID, principalmente em pontos da trajetória sujeitos a incertezas na estimativa dos erros, o que é importante ao emular a metodologia para ambientes reais.

A segunda parte do capítulo visou apresentar testes envolvendo a fusão do algoritmo de detecção de objetos baseado em redes neurais com o controlador adaptativo para testes em linhas de distribuição urbana. Inicialmente, foram feitos estudos para a escolha do modelo de rede a ser utilizado para as missões envolvendo o controle. Foi mostrado que o YOLOv8 apresentou métricas de detecção superiores se comparadas ao YOLOv5, com um custo-benefício maior, levando em conta fatores como detectabilidade e esforço computacional. Análises feitas na máquina utilizada para os testes de campo indicaram que a versão mais leve da YOLOv8 (nano) é a que melhor se encaixa em relação ao requisito temporal do projeto. Com isso, o próximo passo foi testar a detecção das linhas em conjunto com o controlador IT2_PID. Os resultados apresentados mostraram que o drone foi capaz de se alinhar em relação às linhas em um ambiente sujeito a fatores externos, como vento, variabilidade de iluminação e elementos retilíneos no cenário.

Por fim, foram realizados testes aplicando uma velocidade de avanço ao drone, de forma que o mesmo fosse capaz de acompanhar as linhas de distribuição e, durante o seguimento, além de ser capaz de identificar os postes de energia visitados e elementos menores, como as cruzetas e isoladores. A metodologia proposta dá à liberdade ao operador de pausar o seguimento autônomo para verificar algum ponto de interesse de forma manual e, em seguida, retornar ao modo autônomo ou finalizar a missão. Resultados gráficos e visuais mostraram que o drone foi capaz de se manter acima das linhas em trajetórias majoritariamente retilíneas e com variações repentinas na orientação. Na missão de inspeção, após o controle manual, o drone foi deixado em uma posição crítica, longe das linhas. Ao retornar ao modo autônomo, o mesmo foi capaz de corrigir a posição corretamente e seguir até o fim

do trajeto.

5 Conclusões e Propostas Futuras

5.1 Conclusões

O presente trabalho teve como objetivo principal apresentar uma metodologia para a inspeção de linhas de distribuição de energia por meio de uma câmera acoplada a um drone, cobrindo as etapas de identificação dos cabos elétricos e a aplicação de sinais de controle, de forma que o drone mantenha-se posicionado em uma posição e orientação estipuladas pelo operador. O sistema proposto cobre as etapas de aquisição de imagens, identificação das linhas, cálculo da posição do objeto a ser seguido/monitorado e a determinação das ações de controle a serem aplicadas ao drone durante a missão.

A detecção das linhas é realizada por meio do YOLOv8, um identificador de objetos baseado em Redes Neurais Convolucionais. A sua escolha, em detrimento às técnicas convencionais que realizam a identificação dos cabos por meio de reconhecimento de padrões retilíneos nas imagens, se deu devido às possíveis detecções espúrias que podem afetar a movimentação do drone e, por consequência, a qualidade das imagens obtidas durante a missão.

A determinação dos sinais de controle adotados pelo drone, de forma a manter o objeto de interesse visível durante a inspeção, se dá através de uma técnica baseada em informações visuais. Inicialmente, a região coberta pelas linhas foi identificada por meio de um processo de segmentação na imagem e, assim, uma posição única referente à porção central dessa área é determinada. Desta forma, é possível inferir dois erros visuais: um relacionado ao desvio longitudinal da posição do objeto com relação ao referencial e o outro que leva em conta a orientação do objeto na imagem. Estes erros são introduzidos a um controlador PID sintonizado por lógica *fuzzy* do tipo 2. Neste tipo de controle, os ganhos são escolhidos adaptativamente de acordo com a magnitude do erro e seu comportamento temporal. A escolha por essa vertente de controle se deu devido à sua capacidade de lidar com as incertezas no cálculo dos erros, que é um ponto sensível em sistemas baseados em erros visuais.

Os sub-processos relatados acima foram testados com sucesso em um ambi-

ente controlado e, em seguida, validados por meio de imagens reais de cabos de energia obtidas por meio de testes de alinhamento acima das linhas de distribuição, através de um drone físico. O YOLOv8 se sobressaiu em relação ao seu antecessor YOLOv5 com relação às métricas de desempenho, com um leve acréscimo no esforço computacional. Após testes de viabilidade da técnica em uma máquina portátil, foi determinado que o modelo YOLOv8n é o mais adequado para a detecção das linhas em um tempo hábil para sua inserção no sistema completo. Com relação à técnica de controle, o IT2_PID se sobressaiu em relação à sua versão análoga T1_PID, que utiliza a lógica *fuzzy* do tipo 1, bem como abordagens clássicas de controle PID de ganhos fixos. A simulação de uma missão de inspeção mostrou que a vantagem do sistema de controle baseado em lógica *fuzzy* do tipo 2 com relação ao tipo 1 é acentuada nos pontos em que existe um maior grau de incerteza no cálculo da trajetória, o que justifica sua escolha para implementação em ambientes reais.

Os testes de seguimento e inspeção possuem a capacidade de identificação de componentes presentes no sistema de energia, realizados pelo modelo YOLOv8 com a tarefa de detecção de objetos, servindo de alerta para o operador em caso de necessidade de uma inspeção mais minuciosa de algum componente. Na tela de gerenciamento das operações, são mostradas informações úteis, como a posição das linhas, os componentes detectados, velocidades, altitude e o nível de bateria do drone.

A metodologia implementada neste trabalho foi criada integralmente por meio de pacotes de código aberto com algoritmos na linguagem Python. O método de sintonização dos limiares dos ganhos a serem adaptados pelo bloco de lógica *fuzzy* do tipo 2 é facilmente aplicável a diferentes modelos de drones, permitindo que os códigos desenvolvidos neste trabalho sirvam de base para a aplicação do que foi proposto em sistemas mais robustos, com mais sensores e atuadores, permitindo um sensoreamento mais preciso e seguro.

Adicionalmente, destaca-se que o desempenho dos algoritmos de reconhecimento de imagens empregados neste trabalho está diretamente relacionado à qualidade dos dados visuais adquiridos durante a missão. Fatores como a distância entre o drone e os cabos, a resolução das imagens, as condições de iluminação,

o nível de contraste e a presença de ruídos ou oclusões podem impactar significativamente a precisão das etapas de detecção e segmentação. Em particular, variações de luminosidade e condições ambientais adversas podem comprometer a extração de características relevantes, afetando tanto a identificação das linhas quanto a estimação dos erros visuais utilizados no controle. Dessa forma, o sucesso da metodologia proposta está condicionado não apenas à robustez dos algoritmos, mas também à qualidade do processo de aquisição das imagens, o que reforça a importância de estratégias adequadas de planejamento de voo e configuração dos sensores para garantir condições favoráveis de captura.

5.1.1 Propostas Futuras

Como pontos para serem investigados em trabalhos futuros, a primeira sugestão é expandir o sistema de detecção dos elementos presentes no sistema de distribuição, de forma a abranger mais componentes, como espaçadores e transformadores. Além disso, modelos de detecção ou classificação podem ser treinados para a identificação de falhas nesses componentes e, em conjunto com outros sensores acoplados ao drone, possibilitem que os operadores tenham ferramentas para decidir por inspeções individualizadas dos elementos com potenciais falhas e, ao término da operação, possuam relatórios robustos acerca da qualidade do sistema, permitindo manutenções localizadas reduzindo custos operacionais para as companhias de energia.

Outro ponto importante a ser considerado é a utilização das coordenadas geodésicas das linhas como uma ferramenta de localização para a escolha dos pontos de partida e de parada das missões, eliminando a necessidade do ajuste manual para o início da inspeção. Além disso, uma questão a ser considerada é a realização da conexão do robô com a estação de solo por meio de transmissão por radiofrequência, que permita a expansão do que foi proposto para missões de longa distância uma vez que a qualidade da transmissão das imagens, que é um ponto sensível para o funcionamento do sistema de seguimento por caminhos visuais, é afetada pela distância devido a interferências das redes de comunicação wifi.

Outro ponto a ser investigado é o ajuste das funções de pertinência e testes adicionais, de forma a determinar se o aumento do número destas funções, bem como

investigar se a utilização de funções diferentes leva a uma melhora nas métricas de controle, tanto para o estado transiente quanto para o estado estacionário do sistema. Além disso, o conjunto de regras pode ser adaptado, levando em consideração variáveis exógenas como: topologia das linhas, níveis de iluminação, estação do ano ou condição do terreno. Uma outra vertente a ser explorada é a utilização de técnicas de autoaprendizagem ou aprendizagem por reforço, de forma que o modelo de adaptação dos ganhos seja aperfeiçoado continuamente.

Com relação ao bloco de controle, uma opção a ser explorada é a comparação da metodologia proposta com diferentes métodos de sintonização de controles PID, bem como diferentes topologias de controle, visando aperfeiçoar as métricas de desempenho e introduzir características de interesse, como um baixo sobressalto máximo ou pequenas oscilações em pontos com baixos níveis de erro, que possam afetar a qualidade das imagens obtidas durante a inspeção e até a instabilidade do drone.

REFERÊNCIAS

- [Abdelfattah, Wang e Wang 2020]ABDELFATTAH, R.; WANG, X.; WANG, S. Ttpla: An aerial-image dataset for detection and segmentation of transmission towers and power lines. In: *Proceedings of the Asian Conference on Computer Vision*. [S.l.: s.n.], 2020.
- [Adou, Xu e Chen 2019]ADOU, M. W.; XU, H.; CHEN, G. Insulator faults detection based on deep learning. In: IEEE. *2019 IEEE 13th International Conference on Anti-counterfeiting, Security, and Identification (ASID)*. [S.l.], 2019. p. 173–177.
- [Akhyar e Omatu 1993]AKHYAR, S.; OMATU, S. Self-tuning pid control by neural-networks. In: IEEE. *Proceedings of 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan)*. [S.l.], 1993. v. 3, p. 2749–2752.
- [Akinlar e Topal 2011]AKINLAR, C.; TOPAL, C. Edlines: A real-time line segment detector with a false detection control. *Pattern Recognition Letters*, Elsevier, v. 32, n. 13, p. 1633–1642, 2011.
- [Alhassan et al. 2020]ALHASSAN, A. B. et al. Power transmission line inspection robots: A review, trends and challenges for future research. *International Journal of Electrical Power & Energy Systems*, Elsevier, v. 118, p. 105862, 2020.
- [Andrade et al. 2021]ANDRADE, F. A. et al. Unmanned aerial vehicles motion control with fuzzy tuning of cascaded-pid gains. *Machines*, MDPI, v. 10, n. 1, p. 12, 2021.
- [Araar e Aouf 2014]ARAAR, O.; AOUF, N. Visual servoing of a quadrotor uav for autonomous power lines inspection. In: IEEE. *22nd Mediterranean Conference on Control and Automation*. [S.l.], 2014. p. 1418–1424.
- [Arji et al. 2019]ARJI, G. et al. Fuzzy logic approach for infectious disease diagnosis: A methodical evaluation, literature and classification. *Biocybernetics and biomedical engineering*, Elsevier, v. 39, n. 4, p. 937–955, 2019.
- [Ayoub e Schneider-Kamp 2021]AYOUB, N.; SCHNEIDER-KAMP, P. Real-time on-board deep learning fault detection for autonomous uav inspections. *Electronics*, MDPI, v. 10, n. 9, p. 1091, 2021.
- [Azevedo et al. 2019]AZEVEDO, F. et al. Real-time lidar-based power lines detection for unmanned aerial vehicles. In: IEEE. *2019 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. [S.l.], 2019. p. 1–8.

- [Balovsyak et al. 2024]BALOVSYAK, S. et al. Face mask recognition by the viola-jones method using fuzzy logic. *International Journal of Image, Graphics and Signal Processing (IJIGSP)*, v. 16, n. 3, p. 39–51, 2024.
- [Bellou, Pisica e Banitsas 2024]BELLOU, E.; PISICA, I.; BANITSAS, K. Aerial inspection of high-voltage power lines using yolov8 real-time object detector. *Energies*, MDPI, v. 17, n. 11, p. 2535, 2024.
- [Borase et al. 2021]BORASE, R. P. et al. A review of pid control, tuning methods and applications. *International Journal of Dynamics and Control*, Springer, v. 9, p. 818–827, 2021.
- [Campbell et al. 2018]CAMPBELL, S. et al. Sensor technology in autonomous vehicles: A review. In: IEEE. *2018 29th Irish Signals and Systems Conference (ISSC)*. [S.l.], 2018. p. 1–4.
- [Candan et al. 2023]CANDAN, F. et al. Real-time interval type-2 fuzzy control of an unmanned aerial vehicle with flexible cable-connected payload. *Algorithms*, MDPI, v. 16, n. 6, p. 273, 2023.
- [Canny 1986]CANNY, J. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, Ieee, n. 6, p. 679–698, 1986.
- [Carvajal, Chen e Ogmén 2000]CARVAJAL, J.; CHEN, G.; OGMEN, H. Fuzzy pid controller: Design, performance evaluation, and stability analysis. *Information sciences*, Elsevier, v. 123, n. 3-4, p. 249–270, 2000.
- [Carvalho et al. 2021]CARVALHO, G. et al. Hybrid pid-fuzzy controller for autonomous uav stabilization. In: IEEE. *2021 14th IEEE International Conference on Industry Applications (INDUSCON)*. [S.l.], 2021. p. 1296–1302.
- [Castanón-Puga et al. 2015]CASTANÓN-PUGA, M. et al. Hybrid-intelligent mobile indoor location using wi-fi signals. In: *Proceedings of the 17th International Conference on Enterprise Information Systems, Barcelona, Spain*. [S.l.: s.n.], 2015. p. 27–30.
- [Cerón, Mondragón e Prieto 2018]CERÓN, A.; MONDRAGÓN, I.; PRIETO, F. Onboard visual-based navigation system for power line following with uav. *International Journal of Advanced Robotic Systems*, SAGE Publications Sage UK: London, England, v. 15, n. 2, p. 1729881418763452, 2018.
- [Chaumette 2007]CHAUMETTE, F. Potential problems of stability and convergence in image-based and position-based visual servoing. In: *The confluence of vision and control*. [S.l.]: Springer, 2007. p. 66–78.

- [Chen et al. 2021]CHEN, H. et al. Review of agricultural spraying technologies for plant protection using unmanned aerial vehicle (uav). *International Journal of Agricultural and Biological Engineering*, v. 14, n. 1, p. 38–49, 2021.
- [Chen et al. 2016]CHEN, Y. et al. Automatic power line extraction from high resolution remote sensing imagery based on an improved radon transform. *Pattern Recognition*, Elsevier, v. 49, p. 174–186, 2016.
- [Chen et al. 2019]CHEN, Y. et al. Research on uav flight tracking control based on genetic algorithm optimization and improved bp neural network pid control. In: IEEE. *2019 Chinese Automation Congress (CAC)*. [S.l.], 2019. p. 726–731.
- [Chung et al. 2016]CHUNG, C.-L. et al. Detecting bakanae disease in rice seedlings by machine vision. *Computers and electronics in agriculture*, Elsevier, v. 121, p. 404–411, 2016.
- [Corke e Hutchinson 2001]CORKE, P. I.; HUTCHINSON, S. A. A new partitioned approach to image-based visual servo control. *IEEE Transactions on Robotics and Automation*, IEEE, v. 17, n. 4, p. 507–515, 2001.
- [Dai et al. 2016]DAI, J. et al. R-fcn: Object detection via region-based fully convolutional networks. *Advances in neural information processing systems*, v. 29, 2016.
- [Deng et al. 2016]DENG, C. et al. Real time autonomous transmission line following system for quadrotor helicopters. In: IEEE. *2016 International Conference on Smart Grid and Clean Energy Technologies (ICSGCE)*. [S.l.], 2016. p. 61–64.
- [Dev et al. 2020]DEV, M. P. et al. Various tuning and optimization techniques employed in pid controller: A review. In: SPRINGER. *Proceedings of International Conference in Mechanical and Energy Technology: ICMET 2019, India*. [S.l.], 2020. p. 797–805.
- [Dhillon e Verma 2020]DHILLON, A.; VERMA, G. K. Convolutional neural network: a review of models, methodologies and applications to object detection. *Progress in Artificial Intelligence*, Springer, v. 9, n. 2, p. 85–112, 2020.
- [Diniz et al. 2022]DINIZ, L. F. et al. Visual-based assistive method for uav power line inspection and landing. *Journal of Intelligent & Robotic Systems*, Springer, v. 106, n. 2, p. 41, 2022.
- [Dutta et al. 2015]DUTTA, R. et al. Dynamic cattle behavioural classification using supervised ensemble classifiers. *Computers and electronics in agriculture*, Elsevier, v. 111, p. 18–28, 2015.

- [Dwyer et al. 2022]DWYER, B. et al. *Roboflow (version 1.0)[software]*. 2022.
- [Fernandes e Oliveira 2008]FERNANDES, L. A.; OLIVEIRA, M. M. Real-time line detection through an improved hough transform voting scheme. *Pattern recognition*, Elsevier, v. 41, n. 1, p. 299–314, 2008.
- [Gao et al. 2020]GAO, Y. et al. Linespyx: A power line inspection robot based on digital radiography. *IEEE Robotics and Automation Letters*, IEEE, v. 5, n. 3, p. 4759–4765, 2020.
- [Gioi et al. 2012]GIOI, R. G. V. et al. Lsd: A line segment detector. *Image Processing On Line*, v. 2, p. 35–55, 2012.
- [Girshick 2015]GIRSHICK, R. Fast r-cnn. In: *Proceedings of the IEEE international conference on computer vision*. [S.l.: s.n.], 2015. p. 1440–1448.
- [Girshick et al. 2014]GIRSHICK, R. et al. Rich feature hierarchies for accurate object detection and semantic segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2014. p. 580–587.
- [Guan et al. 2021]GUAN, H. et al. Uav-lidar aids automatic intelligent powerline inspection. *International Journal of Electrical Power & Energy Systems*, Elsevier, v. 130, p. 106987, 2021.
- [Gutierrez-Rojas et al. 2022]GUTIERREZ-ROJAS, D. et al. Performance evaluation of machine learning for fault selection in power transmission lines. *Knowledge and Information Systems*, Springer, v. 64, n. 3, p. 859–883, 2022.
- [Han e Jung 2020]HAN, K.; JUNG, H. Trends in logistics delivery services using uav. *Electronics and Telecommunications Trends*, Electronics and Telecommunications Research Institute, v. 35, n. 1, p. 71–79, 2020.
- [Hansen, Pereyra e Scherer 2013]HANSEN, P. C.; PEREYRA, V.; SCHERER, G. *Least squares data fitting with applications*. [S.l.]: JHU Press, 2013.
- [He, Zeng e Hu 2019]HE, T.; ZENG, Y.; HU, Z. Research of multi-rotor uavs detailed autonomous inspection technology of transmission lines based on route planning. *IEEE Access*, IEEE, v. 7, p. 114955–114965, 2019.
- [Helgason e Helgason 1999]HELGASON, S.; HELGASON, S. *The radon transform*. [S.l.]: Springer, 1999. v. 2.
- [Hough 1962]HOUGH, P. V. *Method and means for recognizing complex patterns*. [S.l.]: Google Patents, 1962. US Patent 3,069,654.

- [Hussain 2023]HUSSAIN, M. Yolo-v1 to yolo-v8, the rise of yolo and its complementary nature toward digital manufacturing and industrial defect detection. *Machines*, MDPI, v. 11, n. 7, p. 677, 2023.
- [Jenssen, Roverso et al. 2019]JENSSEN, R.; ROVERSO, D. et al. Intelligent monitoring and inspection of power line components powered by uavs and deep learning. *IEEE Power and energy technology systems journal*, IEEE, v. 6, n. 1, p. 11–21, 2019.
- [Jeong et al. 2018]JEONG, J. et al. Drone-based power-line tracking system. *The Transactions of The Korean Institute of Electrical Engineers*, The Korean Institute of Electrical Engineers, v. 67, n. 6, p. 773–781, 2018.
- [Karnik e Mendel 2001]KARNIK, N. N.; MENDEL, J. M. Centroid of a type-2 fuzzy set. *information SCIences*, Elsevier, v. 132, n. 1-4, p. 195–220, 2001.
- [Ketkar et al. 2021]KETKAR, N. et al. Introduction to pytorch. *Deep Learning with Python: Learn Best Practices of Deep Learning Models with PyTorch*, Springer, p. 27–91, 2021.
- [Ko, Park e Ko 2022]KO, Y. K.; PARK, J. H.; KO, Y. D. A development of optimal algorithm for integrated operation of uavs and uavs for goods delivery at tourist destinations. *Applied Sciences*, MDPI, v. 12, n. 20, p. 10396, 2022.
- [Kosko e Isaka 1993]KOSKO, B.; ISAKA, S. Fuzzy logic. *Scientific American*, JSTOR, v. 269, n. 1, p. 76–81, 1993.
- [Larrauri, Sorrosal e González 2013]LARRAURI, J. I.; SORROSAL, G.; GONZÁLEZ, M. Automatic system for overhead power line inspection using an unmanned aerial vehicle—relifo project. In: IEEE. *2013 International conference on unmanned aircraft systems (ICUAS)*. [S.l.], 2013. p. 244–252.
- [Larson 2016]LARSON, K. Fuzzy logic tuning of a proportional-integral-derivative controller. 2016.
- [Le, Cheng e Liu 2023]LE, M.-H.; CHENG, C.-H.; LIU, D.-G. An efficient adaptive noise removal filter on range images for lidar point clouds. *Electronics*, MDPI, v. 12, n. 9, p. 2150, 2023.
- [Li et al. 2020]LI, J. et al. Deep learning-based bird’s nest detection on transmission lines using uav imagery. *Applied sciences*, MDPI, v. 10, n. 18, p. 6147, 2020.

- [Li et al. 2010]LI, Z. et al. Towards automatic power line detection for a uav surveillance system using pulse coupled neural filter and an improved hough transform. *Machine Vision and Applications*, Springer, v. 21, n. 5, p. 677–686, 2010.
- [Li et al. 2023]LI, Z. et al. Design and application of a uav autonomous inspection system for high-voltage power transmission lines. *Remote Sensing*, MDPI, v. 15, n. 3, p. 865, 2023.
- [Lin et al. 2014]LIN, T.-Y. et al. Microsoft coco: Common objects in context. In: SPRINGER. *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*. [S.l.], 2014. p. 740–755.
- [Liu, Hou e Ju 2017]LIU, X.; HOU, L.; JU, X. A method for detecting power lines in uav aerial images. In: IEEE. *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*. [S.l.], 2017. p. 2132–2136.
- [Lopez et al. 2021]LOPEZ, R. L. et al. Autonomous uav system for cleaning insulators in power line inspection and maintenance. *Sensors*, MDPI, v. 21, n. 24, p. 8488, 2021.
- [Lopez-Sanchez e Moreno-Valenzuela 2023]LOPEZ-SANCHEZ, I.; MORENO-VALENZUELA, J. Pid control of quadrotor uavs: A survey. *Annual Reviews in Control*, Elsevier, v. 56, p. 100900, 2023.
- [Luo, Yu e Yang 2021]LUO, Y.; YU, X.; YANG, D. A new recognition algorithm for high-voltage lines based on improved lsd and convolutional neural networks. *IET Image Processing*, Wiley Online Library, v. 15, n. 1, p. 260–268, 2021.
- [Maharana, Mondal e Nemade 2022]MAHARANA, K.; MONDAL, S.; NEMADE, B. A review: Data pre-processing and data augmentation techniques. *Global Transitions Proceedings*, Elsevier, v. 3, n. 1, p. 91–99, 2022.
- [Mamdani e Assilian 1975]MAMDANI, E. H.; ASSILIAN, S. An experiment in linguistic synthesis with a fuzzy logic controller. *International journal of man-machine studies*, Elsevier, v. 7, n. 1, p. 1–13, 1975.
- [Melo et al. 2022]MELO, A. G. et al. Fuzzy gain-scheduling pid for uav position and altitude controllers. *Sensors*, MDPI, v. 22, n. 6, p. 2173, 2022.
- [Mendel 2013]MENDEL, J. M. General type-2 fuzzy logic systems made simple: a tutorial. *IEEE Transactions on Fuzzy Systems*, IEEE, v. 22, n. 5, p. 1162–1182, 2013.

- [Mendel e John 2002]MENDEL, J. M.; JOHN, R. B. Type-2 fuzzy sets made simple. *IEEE Transactions on fuzzy systems*, IEEE, v. 10, n. 2, p. 117–127, 2002.
- [Menéndez, Pérez e Cheein 2019]MENÉNDEZ, O.; PÉREZ, M.; CHEEIN, F. A. Visual-based positioning of aerial maintenance platforms on overhead transmission lines. *Applied Sciences*, MDPI, v. 9, n. 1, p. 165, 2019.
- [Mirallès et al. 2018]MIRALLÈS, F. et al. Linedrone technology: Landing an unmanned aerial vehicle on a power line. In: IEEE. *2018 IEEE International Conference on Robotics and Automation (ICRA)*. [S.l.], 2018. p. 6545–6552.
- [Mittal et al. 2020]MITTAL, K. et al. A comprehensive review on type 2 fuzzy logic applications: Past, present and future. *Engineering Applications of Artificial Intelligence*, Elsevier, v. 95, p. 103916, 2020.
- [Mohammed et al. 2022]MOHAMMED, M. Q. et al. Review of learning-based robotic manipulation in cluttered environments. *Sensors*, MDPI, v. 22, n. 20, p. 7938, 2022.
- [Mourtzis, Angelopoulos e Panopoulos 2021]MOURTZIS, D.; ANGELOPOULOS, J.; PANOPOULOS, N. Uavs for industrial applications: Identifying challenges and opportunities from the implementation point of view. *Procedia Manufacturing*, Elsevier, v. 55, p. 183–190, 2021.
- [Nawaz et al. 2024]NAWAZ, R. et al. Timeseries fault classification in power transmission lines by non-intrusive feature extraction and selection using supervisare machine learning. *IEEE Access*, IEEE, 2024.
- [Nica, Delcea e Chiriță 2024]NICA, I.; DELCEA, C.; CHIRIȚĂ, N. Mathematical patterns in fuzzy logic and artificial intelligence for financial analysis: a bibliometric study. *Mathematics*, MDPI, v. 12, n. 5, p. 782, 2024.
- [Nise 2020]NISE, N. S. *Control systems engineering*. [S.l.]: John Wiley & Sons, 2020.
- [Oersted e Ma 2023]OERSTED, H.; MA, Y. Review of pid controller applications for uavs. *arXiv preprint arXiv:2311.06809*, 2023.
- [Ogata 2003]OGATA, K. *Ingeniería de control moderna*. [S.l.]: Pearson Educación, 2003.
- [O’Shea e Nash 2015]O’SHEA, K.; NASH, R. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

- [Pantazi, Moshou e Bravo 2016]PANTAZI, X.-E.; MOSHOU, D.; BRAVO, C. Active learning system for weed species recognition based on hyperspectral sensing. *Biosystems Engineering*, Elsevier, v. 146, p. 193–202, 2016.
- [Park et al. 2020]PARK, J.-Y. et al. Method of operating a gis-based autopilot drone to inspect ultrahigh voltage power lines and its field tests. *Journal of Field Robotics*, Wiley Online Library, v. 37, n. 3, p. 345–361, 2020.
- [Pathak, Mukherjee e Misra 2021]PATHAK, N.; MUKHERJEE, A.; MISRA, S. Aerialblocks: Blockchain-enabled uav virtualization for industrial iot. *IEEE Internet of Things Magazine*, IEEE, v. 4, n. 1, p. 72–77, 2021.
- [Perfilieva 2007]PERFILIEVA, I. Analytical theory of fuzzy if-then rules with compositional rule of inference. In: *Fuzzy Logic: A Spectrum of Theoretical & Practical Issues*. [S.l.]: Springer, 2007. p. 174–191.
- [Pinto et al. 2020]PINTO, M. F. et al. A robotic cognitive architecture for slope and dam inspections. *Sensors*, MDPI, v. 20, n. 16, p. 4579, 2020.
- [Pussente et al. 2023]PUSSENTE, G. A. et al. Uav power line tracking control based on a type-2 fuzzy-pid approach. *Robotics*, MDPI, v. 12, n. 2, p. 60, 2023.
- [Qin et al. 2018]QIN, X. et al. Detecting inspection objects of power line from cable inspection robot lidar data. *Sensors*, MDPI, v. 18, n. 4, p. 1284, 2018.
- [Rabah et al. 2019]RABAH, M. et al. Autonomous moving target-tracking for a uav quadcopter based on fuzzy-pi. *IEEE Access*, IEEE, v. 7, p. 38407–38419, 2019.
- [Rafique, Fu e Mai 2021]RAFIQUE, F.; FU, L.; MAI, R. End to end machine learning for fault detection and classification in power transmission lines. *Electric Power Systems Research*, Elsevier, v. 199, p. 107430, 2021.
- [RangeKing 2023]RANGEKING. *Structure of YOLOv8 detection models*. 2023. <https://github.com/ultralytics/ultralytics/issues/189> [Accessed: (November 13, 2023)].
- [Rao et al. 2022]RAO, J. et al. Position control of quadrotor uav based on cascade fuzzy neural network. *Energies*, MDPI, v. 15, n. 5, p. 1763, 2022.
- [Redmon et al. 2016]REDMON, J. et al. You only look once: Unified, real-time object detection. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 779–788.

- [Rosales, Soria e Rossomando 2019] ROSALES, C.; SORIA, C. M.; ROSSOMANDO, F. G. Identification and adaptive pid control of a hexacopter uav based on neural networks. *International journal of Adaptive control and signal processing*, Wiley Online Library, v. 33, n. 1, p. 74–91, 2019.
- [Ross 2005] ROSS, T. J. *Fuzzy logic with engineering applications*. [S.l.]: John Wiley & Sons, 2005.
- [Salhaoui et al. 2019] SALHAOUI, M. et al. Smart industrial iot monitoring and control system based on uav and cloud computing applied to a concrete plant. *Sensors*, MDPI, v. 19, n. 15, p. 3316, 2019.
- [Sarabakha et al. 2017] SARABAKHA, A. et al. Type-2 fuzzy logic controllers made even simpler: From design to deployment for uavs. *IEEE Transactions on Industrial Electronics*, IEEE, v. 65, n. 6, p. 5069–5077, 2017.
- [Saurav et al. 2019] SAURAV, S. et al. Power line segmentation in aerial images using convolutional neural networks. In: SPRINGER. *Pattern Recognition and Machine Intelligence: 8th International Conference, PReMI 2019, Tezpur, India, December 17-20, 2019, Proceedings, Part I*. [S.l.], 2019. p. 623–632.
- [Senthilnath et al. 2016] SENTHILNATH, J. et al. Detection of tomatoes using spectral-spatial methods in remotely sensed rgb images captured by uav. *Biosystems engineering*, Elsevier, v. 146, p. 16–32, 2016.
- [Setyanugraha et al. 2022] SETYANUGRAHA, N. et al. Study of a weather prediction system based on fuzzy logic using mamdani and sugeno methods. *Physics Communication*, v. 6, n. 2, p. 61–70, 2022.
- [Shakiba et al. 2023] SHAKIBA, F. M. et al. Application of machine learning methods in fault detection and classification of power transmission lines: a survey. *Artificial Intelligence Review*, Springer, v. 56, n. 7, p. 5799–5836, 2023.
- [Shinde e Shah 2018] SHINDE, P. P.; SHAH, S. A review of machine learning and deep learning applications. In: IEEE. *2018 Fourth international conference on computing communication control and automation (ICCCUBEA)*. [S.l.], 2018. p. 1–6.
- [Silva 2015] SILVA, A. J. N. d. Análise organizacional de acidentes de trabalho no setor de distribuição de energia elétrica. Universidade Estadual Paulista (UNESP), 2015.
- [Silva et al. 2012] SILVA, M. et al. O setor elétrico brasileiro ea sustentabilidade no século 21: Oportunidades e desafios. *Brasília, 92p* (), Brasília, 92p (<https://www.>

- internationalrivers.org/pt-br/resources/o-setor-el% C3% A9trico-brasileiro-ea-sustentabilidade-no-s% C3% A9culo-21-% E2% 80% 93-oportunidades-e-desafios*). Accessed on, v. 12, n. 07, p. 2016, 2012.
- [Sohn, Jwa e Kim 2012]SOHN, G.; JWA, Y.; KIM, H. B. Automatic powerline scene classification and reconstruction using airborne lidar data. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Copernicus GmbH, v. 1, p. 167–172, 2012.
- [Solilo e Doorsamy 2018]SOLILO, M.; DOORSAMY, W. Power line tracking unmanned arial vehicle (uav). In: *Proceedings of the internacional conference on industrial engineering and operafions management*. [S.l.: s.n.], 2018.
- [Son et al. 2023]SON, H.-S. et al. Recognition of the shape and location of multiple power lines based on deep learning with post-processing. *IEEE Access*, IEEE, 2023.
- [Soori, Arezoo e Dastres 2023]SOORI, M.; AREZOO, B.; DASTRES, R. Artificial intelligence, machine learning and deep learning in advanced robotics, a review. *Cognitive Robotics*, Elsevier, 2023.
- [Stambler, Sherwin e Rowe 2019]STAMBLER, A.; SHERWIN, G.; ROWE, P. Detection and reconstruction of wires using cameras for aircraft safety systems. In: IEEE. *2019 International Conference on Robotics and Automation (ICRA)*. [S.l.], 2019. p. 697–703.
- [Szeliski 2022]SZELISKI, R. *Computer vision: algorithms and applications*. [S.l.]: Springer Nature, 2022.
- [Takaya et al. 2021]TAKAYA, K. et al. Tracking control of unmanned aerial vehicle for power line inspection. In: *Motion Planning*. [S.l.]: IntechOpen, 2021.
- [Teng et al. 2017]TENG, G. et al. Mini-uav lidar for power line inspection. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Copernicus GmbH, v. 42, p. 297–300, 2017.
- [Terven e Cordova-Esparza 2023]TERVEN, J.; CORDOVA-ESPARZA, D. A comprehensive review of yolo: From yolov1 and beyond. arxiv 2023. *arXiv preprint arXiv:2304.00501*, 2023.
- [Tian, Wang e Zhu 2015]TIAN, F.; WANG, Y.; ZHU, L. Power line recognition and tracking method for uavs inspection. In: IEEE. *2015 IEEE international conference on information and automation*. [S.l.], 2015. p. 2136–2141.

- [Uzakov, Nascimento e Saska 2020]UZAKOV, T.; NASCIMENTO, T. P.; SASKA, M. Uav vision-based nonlinear formation control applied to inspection of electrical power lines. In: IEEE. *2020 international conference on unmanned aircraft systems (ICUAS)*. [S.l.], 2020. p. 1301–1308.
- [Valipour, Khandani e Moradi 2015]VALIPOUR, S.; KHANDANI, H.; MORADI, H. The design and implementation of a hotline tracking uav. In: IEEE. *2015 3rd RSI International Conference on Robotics and Mechatronics (ICROM)*. [S.l.], 2015. p. 252–258.
- [Vemula e Frye 2020]VEMULA, S.; FRYE, M. Real-time powerline detection system for an unmanned aircraft system. In: IEEE. *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. [S.l.], 2020. p. 4493–4497.
- [Wang et al. 2023]WANG, G. et al. Uav-yolov8: A small-object-detection model based on improved yolov8 for uav aerial photography scenarios. *Sensors*, MDPI, v. 23, n. 16, p. 7190, 2023.
- [Wang et al. 2020]WANG, G. et al. Field evaluation of spray drift and environmental impact using an agricultural unmanned aerial vehicle (uav) sprayer. *Science of the Total Environment*, Elsevier, v. 737, p. 139793, 2020.
- [Wang et al. 2018]WANG, Y. et al. A hierarchical unsupervised method for power line classification from airborne lidar data. *International journal of digital earth*, Taylor & Francis, 2018.
- [Wang et al. 2015]WANG, Y. et al. Fuzzy radial basis function neural network pid control system for a quadrotor uav based on particle swarm optimization. In: IEEE. *2015 IEEE International Conference on Information and Automation*. [S.l.], 2015. p. 2580–2585.
- [Wu e Mendel 2008]WU, D.; MENDEL, J. M. Enhanced karnik–mendel algorithms. *IEEE transactions on fuzzy systems*, IEEE, v. 17, n. 4, p. 923–934, 2008.
- [Xu, Oja e Kultanen 1990]XU, L.; OJA, E.; KULTANEN, P. A new curve detection method: randomized hough transform (rht). *Pattern recognition letters*, Elsevier, v. 11, n. 5, p. 331–338, 1990.
- [Yan et al. 2007]YAN, G. et al. Automatic extraction of power lines from aerial images. *IEEE Geoscience and remote sensing letters*, IEEE, v. 4, n. 3, p. 387–391, 2007.

- [Yetgin, Senturk e Gerek 2015]YETGIN, O. E.; SENTURK, Z.; GEREK, O. N. A comparison of line detection methods for power line avoidance in aircrafts. In: IEEE. *2015 9th International Conference on Electrical and Electronics Engineering (ELECO)*. [S.l.], 2015. p. 241–245.
- [Yuan et al. 2023]YUAN, J. et al. Identification method of typical defects in transmission lines based on yolov5 object detection algorithm. *Energy Reports*, Elsevier, v. 9, p. 323–332, 2023.
- [Zadeh 1965]ZADEH, L. A. Fuzzy sets. *Information and control*, Elsevier, v. 8, n. 3, p. 338–353, 1965.
- [Zengin et al. 2020]ZENGIN, A. T. et al. Measurement of power line sagging using sensor data of a power line inspection robot. *Ieee Access*, IEEE, v. 8, p. 99198–99204, 2020.
- [Zhang et al. 2019]ZHANG, H. et al. Detecting power lines in uav images with convolutional features and structured constraints. *Remote Sensing*, MDPI, v. 11, n. 11, p. 1342, 2019.
- [Zhang 2002]ZHANG, Z. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, IEEE, v. 22, n. 11, p. 1330–1334, 2002.
- [Zhao et al. 2021]ZHAO, S. et al. Structural health monitoring and inspection of dams based on uav photogrammetry with image 3d reconstruction. *Automation in Construction*, Elsevier, v. 130, p. 103832, 2021.
- [Zhao et al. 2020]ZHAO, T. et al. General type-2 fuzzy gain scheduling pid controller with application to power-line inspection robots. *International Journal of Fuzzy Systems*, Springer, v. 22, p. 181–200, 2020.
- [Zhao, Dong e Zuo 2022]ZHAO, W.; DONG, Q.; ZUO, Z. A method combining line detection and semantic segmentation for power line extraction from unmanned aerial vehicle images. *Remote Sensing*, MDPI, v. 14, n. 6, p. 1367, 2022.
- [Zhou et al. 2016]ZHOU, G. et al. Robust real-time uav based power line detection and tracking. In: IEEE. *2016 IEEE International Conference on Image Processing (ICIP)*. [S.l.], 2016. p. 744–748.