

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
ENGENHARIA ELETRÔNICA E DE TELECOMUNICAÇÕES
CAMPUS PATOS DE MINAS

MARCIO ALEXANDRE JULIETI CROCHELA

**Desenvolvimento de Controle Autônomo para Execução de
Manobras Acrobáticas em Drones Crazyflie Utilizando Processos
Gaussianos**

Patos de Minas - MG
2026

MARCIO ALEXANDRE JULIETI CROCHELA

**Desenvolvimento de Controle Autônomo para Execução de
Manobras Acrobáticas em Drones Crazyflie Utilizando Processos
Gaussianos**

Projeto de pesquisa apresentado à banca examinadora como requisito parcial de avaliação da disciplina de PFC2 da graduação em Engenharia Eletrônica e de Telecomunicações, da Faculdade de Engenharia Elétrica, da Universidade Federal de Uberlândia, Campus Patos de Minas.

Orientador: Prof. Dr. Daniel Costa Ramos

RESUMO

O presente trabalho aborda o desenvolvimento de uma estratégia de controle autônomo para a execução de manobras acrobáticas complexas em nano drones, especificamente a manobra de giro completo para trás no modelo Crazyflie. O objetivo principal do estudo consiste em reproduzir e validar um algoritmo de otimização bayesiana, utilizando modelos substitutos de processos gaussianos, para planejar e executar a acrobacia em um ambiente de simulação de alta fidelidade, empregando controladores clássicos apenas para a estabilização básica de voo. A metodologia adota uma abordagem de aprendizado de máquina onde a manobra é segmentada em cinco fases dinâmicas sequenciais de atuação motora. O ambiente virtual permite que o algoritmo explore e aprenda as incertezas aerodinâmicas por meio de tentativas iterativas. O sistema ajusta autonomamente os tempos de atuação e a força dos motores com base em uma função de custo que avalia a precisão espacial e a estabilidade da recuperação, comparando os achados com um modelo matemático analítico resolvido separadamente. Os resultados demonstram a convergência bem-sucedida do algoritmo de otimização após sete mil iterações de voo, identificando os parâmetros exatos para cada etapa do movimento. A simulação comprova que o veículo executa o giro de trezentos e sessenta graus contido dentro das restrições espaciais estabelecidas, compensando as perdas de sustentação e a turbulência caótica sem a necessidade de modelagem matemática exaustiva das forças de arrasto. Imediatamente após o giro, o controle estabilizador retoma a autoridade mecânica com sucesso, amortecendo a queda livre. Conclui-se que a inteligência artificial, fundamentada em métodos probabilísticos estatísticos, supera as limitações físicas dos modelos lineares tradicionais em cenários de voo agressivos. A arquitetura de controle híbrida desenvolvida mostra-se segura, robusta e eficaz, validando integralmente a viabilidade de transferir futuramente os parâmetros matemáticos otimizados para testes no hardware físico da aeronave.

Palavras-chave: controle autônomo; drones; processos gaussianos; otimização bayesiana; manobras acrobáticas.

ABSTRACT

The present work addresses the development of an autonomous control strategy for executing complex aerobatic maneuvers in nano drones, specifically the full backward rotation maneuver in the Crazyflie model. The main objective of the study consists of reproducing and validating a Bayesian optimization algorithm, using Gaussian process surrogate models, to plan and execute the aerobatic maneuver in a high-fidelity simulation environment, employing classic controllers only for basic flight stabilization. The methodology adopts a machine learning approach where the maneuver is segmented into five sequential dynamic phases of motor actuation. The virtual environment allows the algorithm to explore and learn aerodynamic uncertainties through iterative attempts. The system autonomously adjusts actuation times and motor thrust based on a cost function that evaluates spatial precision and recovery stability, comparing the findings with an analytical mathematical model solved separately. The results demonstrate the successful convergence of the optimization algorithm after seven thousand flight iterations, identifying the exact parameters for each stage of the movement. The simulation proves that the vehicle executes the three hundred and sixty-degree rotation contained within the established spatial constraints, compensating for lift losses and chaotic turbulence without the need for exhaustive mathematical modeling of drag forces. Immediately after the rotation, the stabilizing control successfully resumes mechanical authority, dampening the free fall. It is concluded that artificial intelligence, based on statistical probabilistic methods, overcomes the physical limitations of traditional linear models in aggressive flight scenarios. The developed hybrid control architecture proves to be safe, robust, and effective, fully validating the feasibility of subsequently transferring the optimized mathematical parameters for testing on the physical hardware of the aircraft.

Keywords: autonomous control; Gaussian processes; Bayesian optimization; drones; aerobatic maneuvers.

LISTA DE FIGURAS

Figura 2.1 - Aplicações Comerciais, industriais e militares dos Drones	19
Figura 2.2 - Controle de fluxo dos UAV's.	20
Figura 2.3 - Forças atuantes sobre o drone	21
Figura 2.4 - Drone <i>Crazyflie</i>	23
Figura 4.1 - Gêmeo Digital no Pybullet	39
Figura 5.1 - Resposta Espacial do Veículo X,Y e Z.....	47
Figura 5.2 - Ângulos de Euler (<i>Roll</i> , <i>Pitch</i> e <i>Yaw</i>)	48
Figura 5.3 – Convergência da Otimização Bayesiana – <i>Backflip</i>	49
Figura 5.4 – Sequência 3D do <i>Backflip</i> $t=0,1s$, $t=0,3s$ e $t=0,6s$	52
Figura 5.5 – Comparação de Trajetória: Altitude (Z) e Ângulo da Manobra.....	53
Figura 5.6 – Planejamento de Controle: As 5 Fases do <i>Backflip</i>	54
Figura 5.7 – Comparação de Velocidade Angular e Esforço de Controle	55
Figura 5.8 – Comparação Espacial 2D da Trajetória Acrobática	57
Figura 5.9 – Gráficos de RPM0, RPM1, RPM2 e RPM3.....	58
Figura 5.10 – Gráficos de Velocidades Lineares V_X , V_Y , V_Z e Ângulos Roll e Pitch.....	59
Figura 5.11 – Gráficos de guinada (Yaw) e as Velocidades Angulares (W_R , W_P , W_Y)	59
Figura 5.12 – Gráficos de Posição e Ângulos	60
Figura 5.13 – Decolagem/Início do Salto	61
Figura 5.14 – Decolagem/Início do Salto em 90 graus	61
Figura 5.15– Ponto Intermediário (Ápice do Voo Inercial):	62
Figura 5.16– Ponto Intermediário em 270 graus:	62
Figura 5.17– Ponto Final (Recuperação e Estabilização):.....	63

LISTA DE TABELAS

Tabela 3.1- Materiais utilizados	35
Tabela 4.1- Primitivas do Movimento	42
Tabela 5.1- Parâmetros Otimizados para as Fases do Backflip.....	50

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
ARM	<i>Advanced RISC Machine</i>
CFD	<i>Computational Fluid Dynamics</i>
GP	Processos Gaussianos
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
ISA	<i>The Instrumentation, Systems, and Automation Society</i>
LIDAR	<i>Light Detection and Ranging</i>
LiPo	Polímero de lítio
MV	Variável Manipulada / <i>Manipulated Variable</i>
ODE	<i>Open Dynamics Engine</i>
PID	Proporcional, Integral e Derivativo
PV	Variável de Processo / <i>Process Variable</i>
QP	Programação Quadrática
RPM	Rotações por Minuto
SARP	Sistemas de Aeronaves Remotamente Pilotadas.
SDK	<i>Software Development Kit</i>
SLAM	<i>Simultaneous Localization and Mapping</i>
SP	Ponto de Ajuste / <i>Set Point</i>
UAV	<i>Unmanned Aerial Vehicle</i>
URDF	<i>Universal Robot Description Format</i>
VANTs	Veículos Aéreos Não Tripulados

LISTA DE ABREVIATURAS DE UNIDADES DE MEDIDAS E GRANDEZAS

Hz	Hertz
Kg	Quilogramas
m	Metros
m/s	Metros por segundo
m/s ²	Metros por segundo ao quadrado
mAh	Miliampere-hora
mm	Milímetros
ms	Milissegundos
Pa	Pascal
Rad	Radianos
rad/s	Radianos por segundo
s	Segundos

SUMÁRIO

1	INTRODUÇÃO	12
1.1	TEMA DO PROJETO	14
1.2	PROBLEMATIZAÇÃO	14
1.3	HIPÓTESES	15
1.4	OBJETIVOS	15
1.4.1	Objetivos Gerais	15
1.4.2	Objetivos Específicos.....	16
1.5	JUSTIFICATIVA	16
1.6	CONSIDERAÇÕES FINAIS	18
2	REFERENCIAL TEÓRICO.....	19
2.1	DEFINIÇÃO E CONTEXTUALIZAÇÃO DE DRONES	19
2.2	FUNDAMENTOS DE CONTROLE AUTÔNOMO	21
2.2.1	Princípios Básicos do Controle Autônomo.....	22
2.3	NANO DRONE <i>CRAZYFLIE</i>	23
2.3.1	Aplicações e trabalhos anteriores.....	24
2.4	SISTEMAS DE CONTROLE E PID	25
2.4.1	Controlador PID (Proporcional, Integral e Derivativo)	27
2.5	MODELAGEM MATEMÁTICA DO QUADRICÓPTERO	29
2.5.1	Como o Drone se Localiza (Sistemas de Referência).....	29
2.5.2	Movimento e Rotação	30
2.6	PROCESSOS GAUSSIANOS (GP): O APRENDIZADO DAS INCERTEZAS....	31
2.6.1	Vantagens do uso dos Processos Gaussianos.....	32
2.6.2	Otimização Bayesiana em Malha Aberta	32
2.6.3	Parametrização da Manobra	33
2.7	CONSIDERAÇÕES FINAIS	34
3	METODOLOGIA	35

3.1	MATERIAIS UTILIZADOS.....	35
3.2	ARQUITETURA DO SISTEMA E INTEGRAÇÃO DE SOFTWARE	37
3.3	PROCEDIMENTO EXPERIMENTAL	37
3.3.1	Fase 1: Modelagem Física e Configuração do Gêmeo Digital	37
3.3.2	Fase 2: Implementação do Controle Nominal (PID)	37
3.3.3	Fase 3: Treino da Inteligência Artificial e Otimização Bayesiana	38
3.3.4	Fase 4: Validação Matemática e Análise Comparativa	38
4	DESENVOLVIMENTO	39
4.1	CONFIGURAÇÃO DO AMBIENTE DE SIMULAÇÃO	39
4.1.1	A Importância da Fidelidade Física.....	39
4.1.2	Configuração de Frequências e a Relação de Tempo.....	40
4.2	IMPLEMENTAÇÃO DO CONTROLE NOMINAL (ESTABILIZADOR)	41
4.3	ESTRATÉGIA DA MANOBRA (PRIMITIVAS DE MOVIMENTO)	42
4.4	CICLO DE OTIMIZAÇÃO E FUNÇÃO DE CUSTO	44
4.4.1	Formulação Matemática do Custo.....	44
4.4.2	O Processo Iterativo de Aprendizado	45
4.4.3	O Papel do Processo Gaussiano na Simulação do Quadricóptero	45
5	RESULTADOS.....	47
5.1	ESTABILIZAÇÃO INICIAL E DESEMPENHO DO CONTROLE NOMINAL ..	47
5.2	CONVERGÊNCIA DO TREINAMENTO E OTIMIZAÇÃO BAYESIANA	48
5.3	PARÂMETROS OTIMIZADOS DA MANOBRA	49
5.4	ANÁLISE DA TRAJETÓRIA E VALIDAÇÃO CRUZADA.....	51
5.4.1	Fundamentação do Modelo Matemático Offline	51
5.4.2	Validação Cinemática (Altitude e Atitude)	52
5.4.3	Esforço de Controle e Arquitetura de Estados	54
5.4.4	Confinamento Espacial no Plano 2D	56
5.4.5	Telemetria Completa e Resposta dos Atuadores	58

6	CONCLUSÕES E TRABALHOS FUTUROS	64
	REFERÊNCIAS	65

1 INTRODUÇÃO

Os drones, ou Veículos Aéreos Não Tripulados (VANTs), representam uma categoria de dispositivos aéreos que têm revolucionado significativamente diversas áreas da sociedade contemporânea. Originados inicialmente para aplicações militares, tais máquinas têm se popularizado e diversificado ao longo das últimas décadas, expandindo seu uso para fins comerciais, científicos, recreativos e de serviços. Segundo Michelson e Reace (1998), os VANTs, também conhecidos como Sistemas de Aeronaves Remotamente Pilotadas (SARP), proporcionam uma maneira inovadora e eficiente de interação com o espaço aéreo.

A evolução dos drones é intrinsecamente ligada aos avanços tecnológicos em microeletrônica, que permitiram a miniaturização de componentes como sensores, câmeras, processadores e baterias com maior densidade energética (Michelson e Reace, 1998). Esse progresso facilitou não apenas o desenvolvimento de drones mais acessíveis e versáteis, mas também a ampliação de suas capacidades operacionais, tornando-os capazes de realizar tarefas complexas com precisão e eficiência.

Albiero e Biasi (2016) categorizam os veículos aéreos não tripulados em duas principais classes: os de asa fixa, frequentemente utilizados em contextos militares devido à sua capacidade de longa distância e resistência, e os de asa rotativa, como os multi-rotorés (como os quadricópteros), que se destacam pela sua capacidade de voo estacionário e manobrabilidade, sendo amplamente adotados em aplicações civis e comerciais.

Com a expansão do uso dos drones, suas aplicações têm se diversificado significativamente. Inicialmente empregados em atividades militares como vigilância e reconhecimento, os drones rapidamente encontraram aplicação em áreas como mapeamento geográfico, monitoramento ambiental, entrega de pacotes, filmagem cinematográfica, inspeções de infraestruturas e agricultura de precisão (Albiero & Biasi, 2016). Essa versatilidade tem sido fundamental para impulsionar a eficiência e a inovação em diversos setores, além de contribuir para soluções mais rápidas e sustentáveis.

A importância crescente dos drones é evidente ao considerar seu impacto positivo em áreas como pesquisa científica, gerenciamento de desastres naturais e segurança pública. Esses dispositivos não apenas facilitam o acesso a locais de difícil alcance, mas também proporcionam uma coleta de dados mais precisa e abrangente, contribuindo para tomadas de decisão mais informadas e eficazes.

Com o contínuo avanço tecnológico e o refinamento das regulamentações, espera-se que os drones continuem a desempenhar um papel cada vez mais relevante na sociedade moderna, promovendo a inovação tecnológica e melhorando a qualidade de vida de muitas pessoas ao redor do mundo.

Os drones não apenas se destacam por suas capacidades de coleta de dados e mapeamento, mas também por sua habilidade em realizar manobras acrobáticas complexas, que demonstram a precisão e o controle avançado dessas máquinas. Manobras como *flips*, *loops* e *barrel rolls* são exemplos de movimentos que exigem uma coordenação precisa entre os motores e os sistemas de controle, destacando o potencial dos drones em operações que demandam alta manobrabilidade. Esses movimentos não são apenas demonstrações técnicas, mas também têm aplicações práticas, como em missões de resgate em espaços confinados ou operações de inspeção em locais de difícil acesso.

Particularmente, os nano drones, como o *Crazyflie*, representam um avanço significativo nessa área. Devido ao seu tamanho reduzido e à leveza, esses drones são capazes de realizar manobras acrobáticas em ambientes restritos, onde drones maiores seriam inviáveis. A pesquisa e o desenvolvimento de sistemas de controle para essas manobras em nano drones são fundamentais para expandir as fronteiras do que essas pequenas máquinas podem realizar, tanto em termos de aplicações recreativas quanto em missões críticas, onde a precisão e a agilidade são essenciais.

Embora o *Crazyflie* possua hardware robusto e malhas de controle proporcional, integral e derivado (PID) bem estabelecidas para voo pairado estável, a execução de um *backflip* (giro de 360 graus no eixo de arfagem) exige uma transição rápida entre acelerações máximas e mínimas dos rotores. Essa agressividade gera turbulências e perdas de sustentação que o modelo matemático linear não prevê adequadamente. Portanto, a tentativa de realizar essa manobra puramente por tentativa e erro resulta em altas taxas de queda e na exaustão prematura dos componentes físicos do veículo. Por esse motivo, é necessária uma abordagem preditiva e baseada em dados que consiga aprender a dinâmica complexa da manobra.

Neste contexto, o desenvolvimento e o treinamento dos algoritmos em um ambiente de simulação computacional de alta fidelidade atuam como uma primeira etapa de validação indispensável. O uso do simulador permite explorar as capacidades extremas do nano drone e encontrar os parâmetros ótimos da acrobacia de forma totalmente segura, garantindo que o algoritmo esteja maduro e testado antes de uma futura transferência para o *hardware* real.

1.1 TEMA DO PROJETO

Este trabalho tem como objetivo desenvolver uma estratégia de controle baseada em Otimização Bayesiana e Processos Gaussianos para a execução de manobras acrobáticas (especificamente o *backflip*), utilizando nano-drones. A proposta envolve o uso de algoritmos avançados de planejamento e controle, que permitirão ao drone calcular e ajustar, em tempo real, os movimentos necessários para realizar uma manobra acrobática de forma estável e controlada.

1.2 PROBLEMATIZAÇÃO

A realização de manobras acrobáticas por drones, é um tema de crescente interesse devido aos desafios técnicos e de controle que envolvem. Essas manobras são importantes porque representam o limite da capacidade de controle e manobrabilidade de drones, especialmente em contextos em que a precisão e a agilidade são críticas, como em operações de resgate ou em inspeções em ambientes restritos.

Diferente de voos convencionais, essas manobras exigem uma precisão extrema nos comandos de controle e uma rápida resposta do sistema para evitar falhas que poderiam resultar na queda ou na perda de estabilidade do drone. O desafio é ainda maior quando se considera a miniaturização desses veículos, como os drones *Crazyflie*, onde as limitações de peso, potência e capacidade computacional impõem restrições adicionais ao planejamento e execução de tais manobras.

O problema central a ser enfrentado é como garantir que o drone, ao tentar executar a acrobacia, consiga manter a estabilidade e realizar a manobra de forma segura e precisa. Isso requer o desenvolvimento de algoritmos de controle que possam prever e ajustar os movimentos do drone em tempo real, mesmo diante de perturbações externas ou incertezas nos sensores. A abordagem proposta por Antal, Péni e Tóth (2020), que utiliza processos Gaussianos para o controle e planejamento de manobras acrobáticas, oferece uma solução promissora, mas sua aplicação em plataformas como o *Crazyflie* ainda precisa ser explorada e adaptada.

Trabalhos anteriores, como o de Gageik, Benz e Montenegro (2015), que exploraram o controle de drones de pequeno porte para acrobacias em ambientes confinados, e o de Faessler et al. (2017), que investigaram a execução de manobras agressivas utilizando nano drones, demonstram a viabilidade e os desafios de tais empreendimentos, reforçando a relevância desta pesquisa. Esses estudos destacam a importância de desenvolver sistemas de controle robustos

que possam lidar com as limitações físicas e computacionais dos drones, especialmente em manobras complexas.

Além disso, é necessário considerar as limitações dos sensores embarcados e a capacidade computacional do drone, que podem afetar a precisão dos cálculos e a execução dos comandos. A problematização se concentra, portanto, em como desenvolver um sistema de controle que seja ao mesmo tempo robusto e eficiente, capaz de realizar manobras complexas em um drone de pequeno porte sem comprometer sua integridade ou segurança.

Essa problemática se insere em um contexto mais amplo de pesquisa em robótica aérea, onde o objetivo é aprimorar a autonomia e a capacidade de drones de realizar tarefas complexas de forma autônoma e precisa. A solução deste problema pode não apenas permitir a execução de manobras acrobáticas com drones de pequeno porte, mas também abrir novas possibilidades para o uso de drones em aplicações que exigem alta manobrabilidade e precisão.

1.3 HIPÓTESES

Se aplicada uma arquitetura de controle híbrida, que associa otimização em malha aberta baseada em Processos Gaussianos a um controle PID em malha fechada, será então possível realizar a manobra acrobática de *backflip* (giro completo no eixo de arfagem) de forma estável e precisa. Espera-se que essa abordagem mantenha a integridade e a estabilidade do voo em nano drones do tipo *Crazyflie*, mesmo diante de severas limitações de peso, potência e capacidade computacional. Desta forma, busca-se investigar a eficácia dessa técnica de aprendizado estatístico na execução de dinâmicas complexas, verificando se a calibração preditiva de parâmetros garante a performance desejada especificamente para a categoria de nano drones.

1.4 OBJETIVOS

1.4.1 Objetivos Gerais

O objetivo geral deste trabalho é reproduzir e validar, em ambiente de simulação (*Pybullet*), um algoritmo de Otimização Bayesiana que utiliza modelos substitutos de Processos Gaussianos para planejar e executar a manobra acrobática de *backflip* no nano-drone *Crazyflie* 2.1, utilizando um controlador PID apenas para a estabilização em voo pairado.

1.4.2 Objetivos Específicos

- Analisar as limitações de hardware do drone *Crazyfly*, como capacidade computacional, sensores embarcados e potência, para garantir que o sistema de controle seja compatível com as características físicas e operacionais do drone;
- Modelar a cinemática e a dinâmica de translação e rotação do quadricóptero em um ambiente virtual preciso, ou seja, o *Pybullet*;
- Implementar um controlador PID de atitude e altitude para atuar como estabilizador nominal de base;
- Parametrizar a manobra de *backflip* em fases dinâmicas sequenciais (aceleração, rotação, inércia e recuperação);
- Aplicar a Otimização Bayesiana com Processos Gaussianos para encontrar, iterativamente, os tempos de duração ideais para cada fase da manobra, minimizando o erro de posição ao final da acrobacia.

1.5 JUSTIFICATIVA

A realização de manobras acrobáticas complexas por drones de pequeno porte, como o *Crazyfly*, apresenta desafios técnicos que vão além dos requisitos de voo estável. A capacidade de um drone realizar uma acrobacia com precisão exige um sistema de controle altamente sofisticado, capaz de lidar com a rápida mudança de orientação e velocidade, além de responder a incertezas nos sensores e perturbações externas. Essas características tornam a pesquisa nesta área não apenas desafiadora, mas também extremamente relevante para o avanço das tecnologias de controle autônomo em robótica aérea.

Trabalhos anteriores na área de manobras com drones demonstram a importância e os desafios dessa linha de pesquisa. Por exemplo, Mueggler et al. (2014) investigaram a execução de manobras agressivas em quadricópteros, focando em movimentos rápidos e mudanças abruptas de direção. O estudo mostrou que algoritmos de controle avançados são essenciais para garantir a precisão e a segurança durante essas manobras, especialmente em ambientes complexos e dinâmicos.

Outro estudo relevante é o de Kaufmann et al. (2020), que exploraram a realização de manobras acrobáticas em drones utilizando redes neurais profundas. A pesquisa destacou a complexidade de treinar drones para executar movimentos acrobáticos com alto grau de precisão, abordando a necessidade de sistemas de controle adaptativos que possam responder em tempo real às variáveis do ambiente. Gageik, Benz e Montenegro (2015) também contribuem para esse

campo, investigando a evasão dinâmica de obstáculos por quadricópteros pequenos em ambientes restritos. O estudo enfatiza a necessidade de algoritmos de controle robustos para manter a segurança e a eficácia do drone durante manobras acrobáticas em espaços confinados.

Faessler et al. (2017) abordaram o controle de voo agressivo em quadricópteros usando a teoria da planicidade diferencial. Esse trabalho destaca a importância de métodos de controle precisos para realizar manobras agressivas, como *flips* e *rolls*, de forma estável e segura. Esses trabalhos demonstram que a realização de manobras acrobáticas com drones não é apenas viável, mas também um campo de pesquisa em crescimento, com implicações significativas para a robótica aérea.

O uso de processos Gaussianos para o planejamento e controle dessas manobras, conforme proposto por Antal, Péni e Tóth (2024), oferece uma solução promissora ao permitir que o drone aprenda e adapte-se dinamicamente às variabilidades do ambiente. No entanto, a aplicação dessa abordagem em drones como o *Crazyfly*, que possui limitações de hardware, ainda requer investigação e adaptação. Além disso, a reprodução de técnicas de controle, como as exploradas por Gageik, Benz e Montenegro (2015) e Faessler et al. (2017), no contexto de acrobacias com drones pequenos, oferece uma base sólida para avançar na pesquisa.

A pesquisa se justifica pela necessidade de desenvolver algoritmos de controle que sejam tanto robustos quanto eficientes para drones de pequeno porte, proporcionando não apenas a capacidade de realizar manobras acrobáticas, mas também ampliando suas aplicações em áreas onde alta manobrabilidade e precisão são críticas.

Além disso, a implementação bem-sucedida de tais manobras acrobáticas em drones pequenos pode ter implicações significativas em diversas indústrias. Em setores como a inspeção de infraestruturas, o monitoramento ambiental e a agricultura de precisão, a capacidade de drones realizarem movimentos complexos com precisão poderia aumentar a eficiência das operações, reduzir custos e expandir as possibilidades de uso em ambientes de difícil acesso (Kim & Park, 2020).

Por fim, o desenvolvimento de um sistema de controle para manobras acrobáticas também levanta questões importantes relacionadas à segurança e confiabilidade desses drones. À medida que essas tecnologias avançam, é fundamental estabelecer diretrizes éticas e regulatórias que garantam sua integração segura no espaço aéreo compartilhado, prevenindo acidentes e garantindo a privacidade e segurança dos cidadãos (Smith & Jones, 2022). Portanto, esta pesquisa não apenas contribui para o avanço técnico na área de robótica aérea, mas também para o entendimento dos desafios e oportunidades associados ao uso de drones autônomos em aplicações práticas e potencialmente impactantes.

1.6 CONSIDERAÇÕES FINAIS

Este trabalho visa desenvolver e implementar um sistema de controle no drone *Crazyflie* para realizar a manobra de *backflip* de forma estável e precisa, superando os desafios técnicos impostos pela miniaturização e pelas limitações de hardware. Espera-se que essa pesquisa contribua para o avanço das tecnologias de controle autônomo em robótica aérea, ampliando as possibilidades de aplicação de drones em setores que exigem alta manobrabilidade e precisão.

A estrutura deste trabalho está organizada da seguinte forma: o capítulo 2 aborda o referencial teórico, discutindo temas como a definição e contextualização dos drones, fundamentos de controle autônomo, as características técnicas do nano-drone *Crazyflie* e os sistemas de controle clássicos e avançados, no capítulo 3 é descrita a metodologia utilizada no planejamento do trabalho, incluindo uma análise do hardware e software empregados, com ênfase na utilização do simulador *Pybullet* para a validação dos algoritmos, o capítulo 4 detalha o desenvolvimento do projeto, explicando a construção do ambiente de simulação, a modelagem da dinâmica do drone e a implementação técnica das fases da manobra acrobática, e por fim, o capítulo 5, são apresentados e discutidos os resultados obtidos, acompanhados de gráficos e análises comparativas, encerrando com o capítulo 6 com as conclusões e propostas para futuras pesquisas na área.

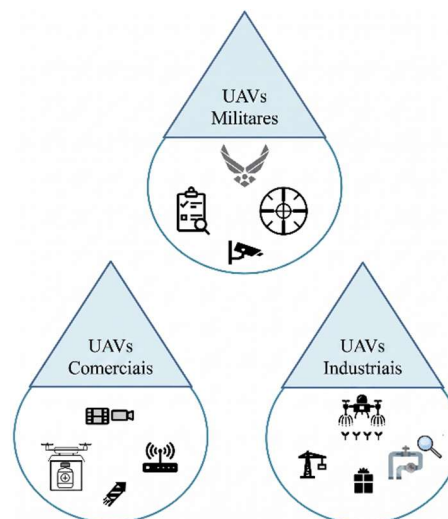
2 REFERENCIAL TEÓRICO

Neste capítulo, são apresentados os conceitos teóricos fundamentais que embasam o desenvolvimento deste projeto. Serão discutidos a definição e a modelagem aerodinâmica dos drones, os princípios do controle autônomo, as características do nano-drone *Crazyfly* e seus módulos sensoriais. Além disso, aborda-se a teoria dos sistemas de controle (PID), a modelagem matemática do quadricóptero e, por fim, o funcionamento dos Processos Gaussianos e da Otimização Bayesiana aplicados à execução de manobras acrobáticas.

2.1 DEFINIÇÃO E CONTEXTUALIZAÇÃO DE DRONES

Os drones, são plataformas aéreas que operam sem a necessidade de um piloto a bordo, sendo controlados remotamente ou de maneira autônoma através de sistemas embarcados. Esses dispositivos têm se tornado essenciais em diversas aplicações, desde operações militares até usos civis e comerciais, como mapeamento geográfico, monitoramento ambiental, entrega de produtos e produção audiovisual. Segundo Michelson e Reace (1998), os drones surgiram inicialmente como uma ferramenta militar, mas sua versatilidade e capacidade de realizar tarefas complexas os tornaram indispensáveis em vários setores. A Figura 2.1 mostra como os drones são divididos na questão de suas aplicações no mercado.

Figura 2.1 - Aplicações Comerciais, industriais e militares dos Drones.

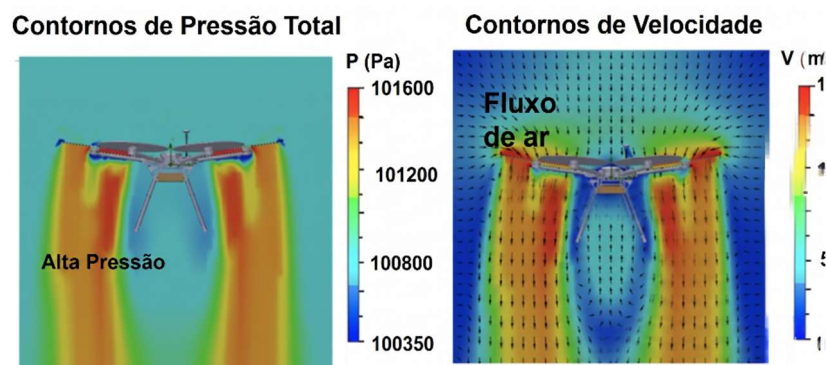


Fonte: Adaptado de Mohsan et al. (2023)

A modelagem de drones, especialmente os de pequeno porte, é um aspecto fundamental para a compreensão e desenvolvimento de suas capacidades. De acordo com Albiero e Biasi (2016), a modelagem matemática de drones inclui a representação de sua dinâmica de voo, que envolve o estudo de forças aerodinâmicas, efeitos de gravidade, e o controle dos motores que

permitem manobras complexas. Essa modelagem é crucial para o desenvolvimento de algoritmos de controle que possibilitam a operação estável e eficiente desses veículos em ambientes variados. A Figura 2.2 mostra como é o controle de fluxo dos drones, mostrando os contornos da pressão total (à esquerda), onde as áreas de alta pressão estão em vermelho, especialmente na abaixo das hélices do drone. Além disso temos os contornos da velocidade (à direita), onde o fluxo de ar se divide e acelera ao redor do drone, com velocidades mais altas em vermelho e mais baixas em azul.

Figura 2.2 - Controle de fluxo dos UAV's.



Fonte: Adaptado de CFD Flow Engineering

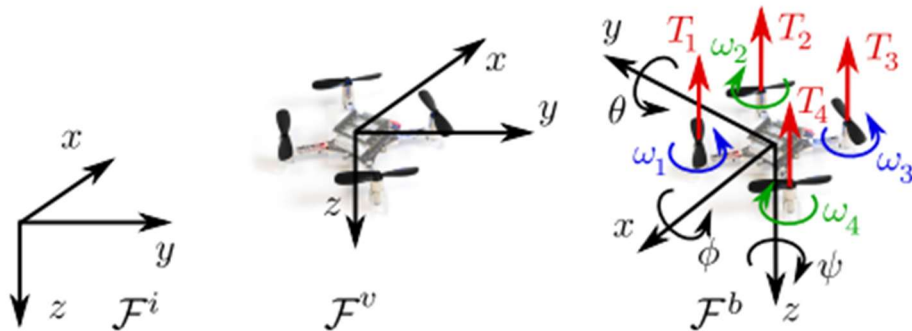
Existem diversas abordagens para a modelagem de drones. Uma das mais comuns é a utilização de modelos baseados em equações diferenciais que descrevem o comportamento dinâmico do drone em termos de suas variáveis de estado, como posição, velocidade, orientação e aceleração (Beard & McLain, 2012). Esses modelos permitem prever o movimento do drone em resposta a comandos de controle, sendo essenciais para a criação de sistemas de navegação e controle.

Outros trabalhos têm explorado a modelagem utilizando técnicas de identificação de sistemas, onde os parâmetros do modelo são ajustados com base em dados experimentais. Essa abordagem é particularmente útil para drones que operam em ambientes dinâmicos ou incertos, onde as condições de voo podem variar significativamente (Bouabdallah, Murrieri & Siegwart, 2004).

Além disso, a miniaturização dos drones, como no caso dos nano drones *Crazyflie*, impõe desafios adicionais à modelagem. Esses drones possuem limitações significativas em termos de peso, potência e capacidade computacional, o que requer modelos simplificados e algoritmos de controle altamente otimizados. Trabalhos como o de Antal, Péni e Tóth (2024) destacam a importância de considerar essas limitações ao desenvolver sistemas de controle para manobras acrobáticas complexas.

A Figura 2.3 mostra a força Inercial $\mathcal{F}^i$, a força Inercial no veículo $\mathcal{F}^v$ e $\mathcal{F}^b$ descrevendo as relações geométricas do veículo e do ambiente como empuxos e velocidades angulares dos rotores, que também são ilustrados.

Figura 2.3 - Forças atuantes sobre o drone



Fonte: Antal, Péni e Tóth (2020)

onde x, y e z são as coordenadas espaciais, T_1, T_2, T_3 e T_4 são as forças de empuxo (tração) individuais geradas por cada um dos quatro rotores, $\omega_1, \omega_2, \omega_3$ e ω_4 são as velocidades angulares de cada rotor, ϕ, θ e ψ são os ângulos de Euler correspondentes, respectivamente, a *Roll* (rolagem ao redor do eixo x_B), *Pitch* (arfagem ao redor do eixo y_B) e *Yaw* (guinada ao redor do eixo z_B).

A modelagem também deve levar em conta os sensores embarcados no drone, como acelerômetros, giroscópios e câmeras, que fornecem os dados necessários para o controle de voo. A fusão de dados desses sensores é um desafio adicional, pois envolve a integração de informações provenientes de diferentes fontes para criar uma estimativa precisa do estado do drone (Mellinger & Kumar, 2011).

Portanto, a definição e modelagem dos drones é uma área de estudo multidisciplinar que combina conhecimentos de aerodinâmica, controle, processamento de sinais e inteligência artificial. Essa base teórica é essencial para o desenvolvimento de sistemas autônomos que possibilitem a execução de tarefas complexas, como manobras acrobáticas, de forma segura e eficiente.

2.2 FUNDAMENTOS DE CONTROLE AUTÔNOMO

O controle autônomo refere-se à capacidade de um sistema, como um drone, de operar de forma independente, sem intervenção humana direta, baseando-se em algoritmos que processam informações de sensores e tomam decisões em tempo real. O desenvolvimento de

controle autônomo é crucial para a evolução dos drones, especialmente quando se trata de tarefas complexas que exigem precisão e rápida resposta, como manobras acrobáticas.

2.2.1 Princípios Básicos do Controle Autônomo

O controle autônomo é geralmente construído sobre a base de três componentes principais: percepção, decisão e ação. A percepção envolve a coleta de dados sensoriais para entender o estado do ambiente e do próprio sistema. A decisão utiliza esses dados para calcular as melhores ações a serem executadas, e a ação é a implementação física dessas decisões.

1. Percepção: A percepção em um drone envolve a utilização de sensores, como câmeras, *Light Detection and Ranging* (LIDAR), acelerômetros, giroscópios e magnetômetros, para medir diversas variáveis, como posição, velocidade, orientação e proximidade de obstáculos. A qualidade do controle autônomo depende diretamente da precisão e confiabilidade dos dados sensoriais (Bouabdallah, Murrieri & Siegwart, 2004);
2. Decisão: Esta fase envolve a análise dos dados coletados para planejar e calcular as ações necessárias para alcançar os objetivos de controle. Algoritmos de planejamento de trajetória, como aqueles baseados em programação dinâmica ou aprendizado de máquina, são usados para calcular a rota ou manobra mais eficiente. Em cenários dinâmicos, onde o ambiente pode mudar rapidamente, a capacidade do drone de ajustar suas decisões em tempo real é essencial;
3. Ação: A ação é a implementação das decisões calculadas, geralmente através de atuadores, como motores e servomecanismos, que controlam o movimento e a orientação do drone. O objetivo do controle é garantir que o drone siga a trajetória planejada ou execute a manobra desejada com a maior precisão possível. Técnicas como controle PID (Proporcional-Integral-Derivativo) são amplamente usadas devido à sua simplicidade e eficiência em sistemas de controle linear.

Um dos principais desafios enfrentados na implementação de controle autônomo é a necessidade de garantir a segurança e confiabilidade do sistema. Isso é especialmente importante em cenários onde os drones operam em proximidade de pessoas ou outras aeronaves. Protocolos de segurança e redundância no controle são frequentemente implementados para mitigar riscos e evitar falhas catastróficas (Shakhatreh et al., 2019).

Outro desafio é a limitação de recursos em drones de pequeno porte, como os nano drones *Crazyfly*. A capacidade computacional e energética limitada exige que os algoritmos de controle sejam altamente eficientes e otimizados para funcionar dentro dessas restrições. Isso

torna o desenvolvimento de controle autônomo para esses drones uma área ativa e desafiadora de pesquisa.

Portanto, os fundamentos do controle autônomo em drones envolvem uma combinação de teoria de controle, inteligência artificial e engenharia de sistemas, com o objetivo de criar plataformas aéreas que possam operar de maneira independente e eficaz em uma ampla gama de aplicações.

2.3 NANO DRONE *CRAZYFLIE*

O *Crazyflie* é um nano drone desenvolvido pela *Bitcraze*, amplamente reconhecido por seu design compacto e modularidade, o que o torna uma ferramenta valiosa tanto para fins educacionais quanto para pesquisas avançadas em robótica e controle autônomo. Com um peso de aproximadamente 27 gramas e dimensões de 92x92x29 mm, o *Crazyflie 2.1* é uma plataforma de código aberto que oferece uma grande flexibilidade para desenvolvedores e pesquisadores. A Figura 2.4 mostra como é o drone *Crazyflie*.

Figura 2.4 - Drone *Crazyflie*



Fonte: *Bitcraze*.

O drone é alimentado por uma bateria de polímero de lítio (LiPo) de 240 mAh, que oferece cerca de sete minutos de tempo de voo contínuo. Ele é equipado com um microcontrolador ARM Cortex-M4, que gerencia as operações de voo e o processamento de dados dos sensores. O controle de voo é suportado por um conjunto de sensores de movimento, incluindo um acelerômetro de 3 eixos, um giroscópio de 3 eixos, um magnetômetro de 3 eixos e um barômetro. Esses sensores trabalham em conjunto para fornecer dados em tempo real sobre a orientação, velocidade angular e altitude do drone.

Um dos aspectos mais notáveis dele é sua capacidade de comunicação sem fio. Ele utiliza a tecnologia de rádio em 2,4 GHz para comunicação com um computador ou dispositivo

móvel, permitindo o controle remoto e a transmissão de dados em tempo real. Essa comunicação é suportada por um sistema de rádio de longo alcance que pode operar em distâncias de até 1 km, dependendo das condições ambientais.

O drone também suporta uma interface *Bluetooth*, o que permite que ele seja controlado por smartphones ou tablets, aumentando ainda mais sua acessibilidade para experimentos educacionais e prototipagem rápida. Além disso, o *Crazyflie* pode ser programado para realizar voos autônomos utilizando software personalizado, que pode ser carregado diretamente no microcontrolador.

2.3.1 Aplicações e trabalhos anteriores

Devido à sua modularidade e facilidade de programação, o *Crazyflie* tem sido amplamente utilizado em pesquisas acadêmicas. Giernacki et al. (2017) realizaram uma revisão abrangente das aplicações do *Crazyflie* em ambientes educacionais e de pesquisa, destacando sua utilização em estudos de controle de voo, navegação autônoma, e aprendizado de máquina. Eles enfatizam que a capacidade do drone de ser equipado com diferentes módulos e sensores o torna uma plataforma ideal para testar novas técnicas de controle e algoritmos de navegação.

Em outro estudo, Palossi et al. (2019) exploraram o uso de aprendizado por reforço profundo para a navegação autônoma do *Crazyflie*. Eles demonstraram como o drone pode ser treinado para realizar tarefas complexas de navegação em ambientes dinâmicos, adaptando-se a mudanças nas condições do ambiente em tempo real. Este trabalho exemplifica o potencial do *Crazyflie* como uma ferramenta para a pesquisa em inteligência artificial e robótica.

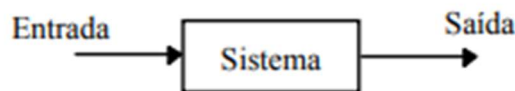
Além de ser usado em pesquisas de ponta, o *Crazyflie* também é uma excelente plataforma para educação em robótica. Estudantes podem aprender sobre os princípios básicos de controle de voo, programação de drones, e integração de sensores, tudo em uma plataforma acessível e fácil de usar. A comunidade de código aberto em torno do nano-drone oferece uma riqueza de recursos e suporte para desenvolvedores e pesquisadores, o que contribui para a contínua expansão de suas capacidades e aplicações.

Essas características tornam o *Crazyflie* uma escolha popular para experimentos que exigem alta manobrabilidade e controle preciso, especialmente em espaços confinados ou ambientes internos. A modularidade dele também permite a fácil adaptação e personalização para uma ampla gama de aplicações, desde experimentos de pesquisa até protótipos de produtos.

2.4 SISTEMAS DE CONTROLE E PID

Os sistemas de controle desempenham um papel fundamental em uma vasta gama de aplicações de engenharia, desde a estabilização de estruturas até o controle de processos industriais e, mais recentemente, no gerenciamento de veículos autônomos, como drones. A essência de um sistema de controle reside na sua capacidade de regular a operação de um sistema físico para alcançar um comportamento desejado, mesmo diante de perturbações externas e incertezas internas, em suma um sistema é uma disposição, conjunto ou coleção de partes conectadas ou relacionadas de tal maneira a formarem um todo. Pode ser físico, biológico, econômico, entre outros, a Figura 2.5 ilustra o sistema:

Figura 2.5 - Diagrama de blocos de um sistema



Fonte: SILVA, Valder Steffen da; MENEGHETTI, Renzo (2003).

Um sistema de controle pode ser definido como um conjunto de dispositivos e algoritmos projetados para regular o comportamento de um sistema dinâmico, fazendo com que ele siga um determinado padrão ou mantenha uma condição estável. Os controladores são a peça central desse sistema, processando informações de sensores e atuando sobre atuadores para manter o sistema em um estado desejado. A importância dos sistemas de controle é amplamente reconhecida em diversas áreas, como na aviação, automação industrial, robótica, e, particularmente, na operação de veículos aéreos não tripulados (UAVs), como os drones. Segundo Ogata (2010), "a teoria de controle permite que sistemas dinâmicos complexos sejam projetados e gerenciados de maneira eficiente, garantindo desempenho, segurança e confiabilidade em operações críticas".

No contexto dos drones, os sistemas de controle são essenciais para garantir a estabilidade e a precisão do voo. Drones como o *Crazyflie*, por exemplo, dependem de sistemas de controle sofisticados para realizar manobras complexas e manter a estabilidade durante o voo.

Estes sistemas garantem que o drone possa seguir uma trajetória desejada, reagir a perturbações externas, e corrigir sua orientação automaticamente em tempo real. Como discutido por Bouabdallah et al. (2004), o controle de drones envolve desafios únicos, como a

necessidade de uma resposta rápida e precisa a mudanças na dinâmica do voo e a capacidade de operar em ambientes altamente não lineares e incertos.

Os sistemas de controle podem ser classificados de várias maneiras, dependendo da abordagem utilizada e das características do sistema a ser controlado. Duas das principais categorias incluem sistemas de controle de malha aberta e malha fechada:

1. **Sistemas de Controle de Malha Aberta:** Nesses sistemas, a saída não influencia as entradas, ou seja, as ações de controle são aplicadas sem considerar o efeito que essas ações têm sobre o sistema. Este tipo de controle é simples, mas pode ser inadequado para sistemas onde as condições variam rapidamente ou onde a precisão é crucial, a Figura 2.6 ilustra um diagrama de blocos desse tipo de sistema:

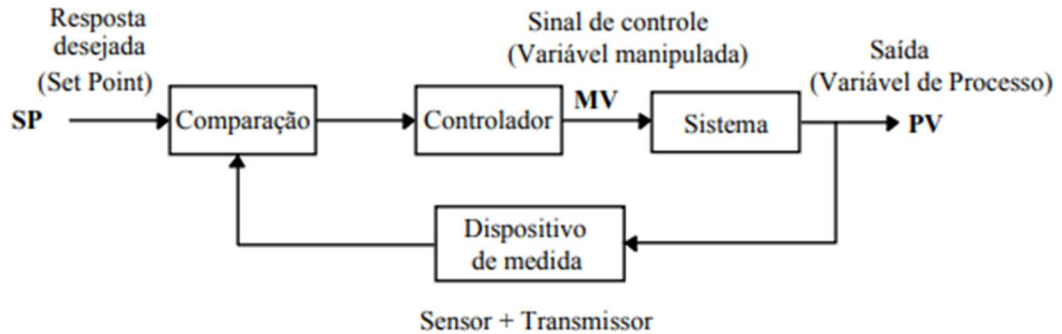
Figura 2.6 - Diagrama de Blocos Malha Aberta



Fonte: SILVA, Valder Steffen da; MENEGHETTI, Renzo (2003).

2. **Sistemas de Controle de Malha Fechada (ou Feedback):** Nestes sistemas, a saída é continuamente monitorada e usada para ajustar as entradas. Este tipo de controle é mais robusto e capaz de lidar com incertezas e perturbações, sendo a base para o controle de drones. Como ressaltado por Dorf e Bishop (2016), "o controle de malha fechada é vital para manter a estabilidade e alcançar o desempenho desejado em sistemas dinâmicos sujeitos a distúrbios imprevisíveis". A Figura 2.7 mostra um diagrama de blocos desse tipo de sistema:

Figura 2.7 - Diagrama de Blocos Malha Fechada



Fonte: SILVA, Valder Steffen da; MENEGHETTI, Renzo (2003).

2.4.1 Controlador PID (Proporcional, Integral e Derivativo)

O controlador PID (Proporcional-Integral-Derivativo) é uma das técnicas de controle mais amplamente utilizadas em sistemas dinâmicos, incluindo drones. A popularidade do PID se deve à sua simplicidade, facilidade de implementação e eficiência em uma ampla gama de aplicações. Este controlador é composto por três componentes principais: o controle proporcional (P), o controle integral (I) e o controle derivativo (D), que, juntos, permitem ao sistema responder de forma rápida e precisa a desvios da condição desejada.

O componente proporcional do controlador PID é responsável por gerar uma correção que é proporcional ao erro instantâneo, ou seja, a diferença entre o valor desejado (*setpoint*) e o valor medido (saída do sistema). Essa correção visa reduzir o erro de forma imediata.

Em termos matemáticos, a ação proporcional é dada por:

$$u(t) = K_p * e(t), \quad (2.1)$$

onde K_p é o ganho proporcional, $e(t)$ é o erro no tempo t .

Assim um ganho proporcional maior tende a diminuir o erro mais rapidamente, mas também pode causar oscilações e instabilidade se for excessivo.

O componente integral acumula o erro ao longo do tempo, corrigindo os desvios sistemáticos que podem não ser eliminados apenas pela ação proporcional. O termo integral busca eliminar o erro residual ao longo do tempo, resultando em uma precisão maior.

A ação integral é dada por:

$$u(t) = K_i * \int e(t) dt, \quad (2.2)$$

onde K_i é o ganho integral.

Embora o termo integral ajude a eliminar o erro estacionário, um valor muito alto de K_i pode causar *overshoot* e instabilidade, pois o sistema pode reagir de forma exagerada aos erros acumulados.

O componente derivativo do controlador PID antecipa o comportamento futuro do erro, aplicando uma correção proporcional à taxa de mudança do erro. Isso ajuda a suavizar a resposta do sistema e reduzir oscilações, melhorando a estabilidade.

A ação derivativa é dada por:

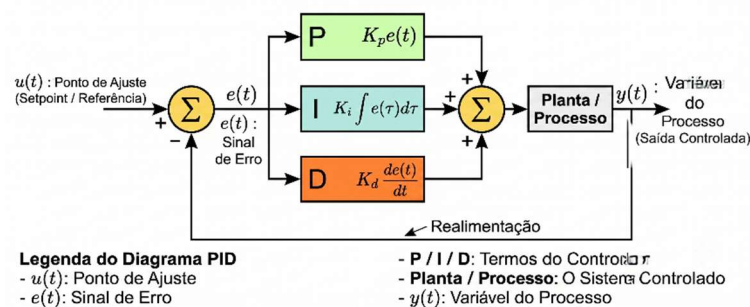
$$u(t) = K_d * \frac{de(t)}{dt}, \quad (2.3)$$

onde K_d é o ganho derivativo.

O termo derivativo é particularmente útil em sistemas onde é necessário amortecer a resposta e evitar oscilações indesejadas. No entanto, o ganho derivativo excessivo pode amplificar o ruído do sistema, resultando em uma resposta irregular.

Como visto anteriormente, o controlador PID emprega três modos, ou seja, controle proporcional, integrador e derivativo. A Figura 2.8 ilustra o diagrama de blocos desse controlador:

Figura 2.8 - Diagrama de Blocos Controlador PID



Fonte: EMBARCADOS. *PID digital (modificado)*

Uma das grandes vantagens do controlador PID é a flexibilidade que ele oferece ao engenheiro ou projetista para ajustar os parâmetros de acordo com as necessidades específicas do sistema em questão. Essa liberdade permite que o controlador seja adaptado a uma vasta gama de aplicações, desde sistemas industriais de grande escala até sistemas de controle mais sensíveis, como o controle de voo em drones.

Os três parâmetros principais do PID, os ganhos proporcional (K_p), integral (K_i) e derivativo (K_d), podem ser ajustados para otimizar o desempenho do sistema, melhorando a estabilidade, a velocidade de resposta e a precisão. Essa capacidade de ajuste é essencial para adaptar o controlador a diferentes condições operacionais e objetivos de controle. Como ressaltado por Astrom e Hagglund (2006), a "liberdade de ajustar os parâmetros do PID permite aos engenheiros equilibrar trade-offs entre rapidez de resposta, estabilidade e robustez".

O controlador PID no tempo contínuo é dado por:

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt}, \quad (2.4)$$

onde $u(t)$ é o sinal de controle, $e(t)$ é o erro no tempo t e as constantes K_p , K_i e K_d são, respectivamente os ganhos proporcional, integral e derivado do controlador PID.

Em sistemas digitais ou de controle discreto, o controlador PID é frequentemente implementado em sua forma discreta. A fórmula discreta do PID, que é derivada da formulação contínua usando métodos como a aproximação da diferença finita, é dada por:

$$u(k) = K_p e(k) + K_i T_s \sum_{i=0}^k e(i) + \frac{K_d}{T_s} (e(k) - e(k-1)). \quad (2.5)$$

onde $u(k)$ é o sinal de controle no instante k , $e(k)$ é o erro no instante k e a constante $T_s \in \mathbb{R}$ é o tempo de amostragem do sistema digital embarcado.

O ajuste dos parâmetros K_p , K_i e K_d em sistemas discretos segue princípios semelhantes aos do sistema contínuo, mas deve considerar a quantização e os atrasos de amostragem. Esses parâmetros podem ser ajustados de forma iterativa ou através de métodos de sintonia como o método de *Ziegler-Nichols*, que fornece regras práticas para determinar valores iniciais dos ganhos baseados em testes de resposta do sistema.

2.5 MODELAGEM MATEMÁTICA DO QUADRICÓPTERO

Para que algoritmos avançados de aprendizado de máquina, como Processos Gaussianos, possam prever e otimizar trajetórias, é fundamental estabelecer primeiro o modelo físico e matemático nominal do sistema. Este modelo descreve como as forças e momentos gerados pelos motores afetam a posição e a orientação do drone *Crazyflie* no espaço.

2.5.1 Como o Drone se Localiza (Sistemas de Referência)

Antes de calcular qualquer movimento, é preciso definir um ponto de vista. Na robótica aérea, é trabalhado com dois pontos de vista (ou sistemas de coordenadas) que conversam entre si o tempo todo:

1. O Sistema do Mundo (Inercial): É um ponto fixo no chão. Nele, o eixo Z sempre aponta para o céu, e os eixos X e Y representam o chão (Norte, Sul, Leste, Oeste);
2. O Sistema do Corpo: É o ponto de vista "pelos olhos do drone". Se o drone inclinar para frente, o eixo Z dele (que antes apontava para o céu) agora apontará na diagonal.

Para o drone navegar, o computador de bordo usa ferramentas matemáticas (como matrizes de rotação ou quatérnios) para traduzir o que é "frente" para o drone em relação ao que

é "Norte" para o mundo. Em manobras acrobáticas como o *backflip*, o uso de quatérnios é preferido, pois evita um erro matemático conhecido como *Gimbal Lock*, que faz o sistema travar quando o drone fica exatamente de cabeça para baixo.

2.5.2 Movimento e Rotação

O voo de um quadricóptero baseia-se na oposição contínua à força gravitacional. Para obter a sustentação necessária, os quatro rotores geram um fluxo de ar descendente que produz, por consequência, uma força de empuxo ascendente.

A Dinâmica de Translação (o movimento de ir para cima, para baixo ou para os lados) depende dessa força. Se o empuxo dos quatro motores for exatamente igual ao peso do drone, ele fica parado no ar (*hovering*). Se o drone se inclina, parte desse empuxo que ia para cima passa a ir para trás, empurrando o drone para frente. A equação física que resume essa segunda lei de Newton aplicada ao drone é:

$$m\ddot{\mathbf{p}} = -mg\hat{\mathbf{e}}_3 + R(f_T\hat{\mathbf{e}}_3) + F_{arrasto} \quad (2.6)$$

onde m é a massa total do drone, $\ddot{\mathbf{p}}$ é a aceleração linear tridimensional do drone, g é a aceleração da gravidade, $\hat{\mathbf{e}}_3$ é o vetor unitário que representa o eixo vertical Z, R é a matriz de rotação, responsável por traduzir a direção do empuxo do corpo do drone para o sistema inercial, f_T é a força de empuxo total e $F_{arrasto}$ são as forças aerodinâmicas externas de resistência do ar que atuam contra o drone.

A dinâmica de rotação descreve a variação da atitude do quadricóptero em torno de seu centro de massa, compreendendo os movimentos de rolagem (*roll*), arfagem (*pitch*) e guinada (*yaw*). Para a execução da manobra de *backflip*, o grau de liberdade preponderante é a arfagem. A indução deste movimento exige a aplicação de um torque diferencial elevado, o qual é obtido por meio de uma variação abrupta no empuxo dos rotores: os motores frontais são submetidos a uma aceleração máxima, enquanto a rotação dos motores traseiros é drasticamente reduzida. Esse gradiente de forças gera o momento angular necessário para rotacionar o veículo em torno de seu eixo transversal. A equação matemática de Euler que descreve essa rotação é:

$$J\dot{\omega} = \tau - \omega \times (J\omega) \quad (2.7)$$

onde J é a matriz de inércia do drone, que representa a resistência do seu corpo em começar ou parar de girar, $\dot{\omega}$ é a aceleração angular, ou seja, o quão rápido o drone altera a sua velocidade de giro, τ é o vetor de torques aplicados pelos motores, ω é a velocidade angular atual em que o drone se encontra girando e a relação $\omega \times (J\omega)$ representa os efeitos giroscópicos, forças que surgem naturalmente como consequência de um objeto já estar em rotação.

Todas essas forças são calculadas com base no design físico do *Crazyflie*. Sob condições normais, como decolar e voar suavemente pelo ambiente, esse modelo matemático exato é suficiente para que um controlador tradicional (como o PID) mantenha o drone estável, pois a variável $F_{arrasto}$ permanece pequena e previsível.

No entanto, quando o drone executa um giro agressivo de 360 graus em uma fração de segundo, ele cria turbulências severas. As hélices passam a engolir o próprio ar agitado que acabaram de empurrar. Isso faz com que a variável de arrasto $F_{arrasto}$ e os efeitos giroscópicos assumam comportamentos altamente não-lineares e caóticos.

Esses efeitos invisíveis fazem as equações matemáticas clássicas falharem. Se tentarmos usar apenas a matemática pura descrita acima acoplada ao PID para dar o *flip*, o drone não conseguirá compensar as perturbações, perderá altitude e cairá. Para que a manobra seja bem-sucedida, precisamos de um sistema capaz de "aprender" com essas turbulências. É para contornar essa falha do modelo matemático estático que introduzimos o aprendizado de máquina, especificamente os Processos Gaussianos.

2.6 PROCESSOS GAUSSIANOS (GP): O APRENDIZADO DAS INCERTEZAS

Como discutido anteriormente, o modelo matemático tradicional tem dificuldades em prever as forças caóticas do ar durante uma manobra acrobática. Para lidar com essa "falha" da física teórica, utilizamos uma ferramenta da estatística e do aprendizado de máquina chamada Processos Gaussianos.

Em vez de tentar calcular a física exata da turbulência (o que seria matematicamente exaustivo), o GP observa o que aconteceu em tentativas passadas do drone e cria uma estimativa probabilística do que deve acontecer na próxima vez (Antal et al., 2024).

Matematicamente, um Processo Gaussiano não retorna apenas um número único como resposta, mas sim uma distribuição de funções. Isso significa que, para cada ponto da manobra, o GP nos dá um valor médio (o que ele acha que vai acontecer) e um grau de incerteza (Antal, 2021).

A definição matemática de um Processo Gaussiano é escrita da seguinte forma:

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')) \quad (2.8)$$

onde $f(x)$ representa a função real que se está tentando aprender (forças invisíveis que atuam no drone), $\mathcal{GP}$ é o símbolo que indica que a função segue a lógica de um Processo Gaussiano, $m(x)$ é a Função Média (representa o "palpite principal" do sistema sobre o comportamento do drone) e $k(x, x')$ é a Função de Covariância ou Kernel, ela define como um

ponto da manobra se relaciona com o outro (se o drone balançou agora, o quão provável é que ele continue balançando no próximo milissegundo).

2.6.1 Vantagens do uso dos Processos Gaussianos

A grande vantagem do uso de Processos Gaussianos em manobras agressivas é a sua capacidade de modelar o erro. Segundo Antal et al. (2024), o GP atua como um "modelo substituto" (*surrogate model*).

Supondo que o drone tenta dar o mortal e cai um pouco para a esquerda, o GP registra esse erro, na próxima tentativa, ele usa a sua função de média para sugerir um pequeno ajuste na força dos motores e usa a sua função de covariância (Kernel) para entender que aquele erro não foi um acaso, mas sim uma característica da aerodinâmica naquele momento específico do giro.

Ao contrário de modelos de aprendizado profundo que exigem bases de dados extensas, os Processos Gaussianos destacam-se pela eficiência com amostras reduzidas (Antal, 2021). Essa característica é fundamental para a viabilidade desta pesquisa, pois permite que o modelo probabilístico convirja para uma solução no simulador *Pybullet* com um número consideravelmente menor de iterações, otimizando o custo computacional na busca pelos parâmetros ideais de voo.

2.6.2 Otimização Bayesiana em Malha Aberta

Com o modelo probabilístico de Processo Gaussiano (GP) estabelecido para prever as incertezas aerodinâmicas, o sistema necessita de uma estratégia matemática para explorar essas previsões e encontrar os comandos ideais de voo. Para cumprir essa tarefa, a presente pesquisa adota a Otimização Bayesiana.

A otimização tradicional, feita por tentativa e erro ou varredura de grade (*grid search*), testa inúmeras combinações de parâmetros até achar a melhor. Em robótica, essa abordagem é inviável, pois cada "teste" exige rodar uma simulação física completa (ou um voo real), o que demanda tempo e alto custo computacional. A Otimização Bayesiana resolve esse problema ao assumir que o cálculo real é "caro" e, portanto, o número de tentativas deve ser o menor possível (Frazier, 2018).

Para alcançar essa eficiência, o algoritmo de Otimização Bayesiana consulta o GP para estimar matematicamente onde a melhor solução de voo provavelmente se encontra, realizando o teste prático apenas nas coordenadas mais promissoras (Antal et al., 2024).

2.6.3 Parametrização da Manobra

Para que o algoritmo consiga otimizar um *backflip*, o movimento não pode ser tratado pelo computador como uma ação única, mas sim traduzido em variáveis ajustáveis. Antal et al. (2024) propõem a divisão da acrobacia em primitivas de movimento, caracterizadas por cinco fases sequenciais em malha aberta.

1. Aceleração (*Accelerate*): Aumento simultâneo do empuxo nos quatro motores para ganhar altitude inicial e compensar a perda de sustentação que ocorrerá durante o giro;
2. Início da Rotação (*Start Rotation*): Aplicação de torque máximo no eixo de *pitch* (diferença de empuxo entre os motores frontais e traseiros) para iniciar a aceleração angular;
3. Voo Inercial (*Coast*): Redução do empuxo geral. O drone continua girando aproveitando a inércia angular adquirida na fase anterior, minimizando o arrasto aerodinâmico adverso quando está invertido;
4. Frenagem da Rotação (*Stop Rotation*): Aplicação de torque reverso para anular a velocidade angular no momento exato em que o drone se aproxima da atitude horizontal (completando os 360 graus);
5. Recuperação (*Recover*): Reativação do empuxo nominal para frear a queda livre e devolver o controle ao sistema de estabilização PID, retomando o voo pairado (*hovering*).

Cada uma dessas fases possui uma duração de tempo específica (t_1, t_2, t_3, t_4, t_5) e um nível de força associado aos motores. O conjunto dessas frações de segundo compõe o vetor de parâmetros exato que a Otimização Bayesiana precisa resolver.

O objetivo final da otimização é encontrar os tempos perfeitos para as cinco fases descritas acima, garantindo que o drone termine o *flip* na mesma posição espacial em que começou assim caracterizando, o erro mínimo.

O mecanismo interno que dita qual será o próximo teste do algoritmo é chamado de Função de Aquisição. Esta função matemática avalia as previsões dadas pelo Processo Gaussiano e decide os parâmetros da próxima tentativa de voo, equilibrando dois conceitos fundamentais durante a busca (Brochu et al., 2010):

1. Exploração (*Exploration*): Testar tempos de motorização em áreas completamente desconhecidas da manobra, onde a incerteza estatística do GP ainda é alta;

2. Aproveitamento (*Exploitation*): Testar tempos muito próximos àqueles que já apresentaram bons resultados nos testes anteriores, buscando refinar a precisão para encontrar a manobra perfeita.

Dessa forma, a arquitetura de otimização opera em um ciclo fechado de aprendizado: a Função de Aquisição sugere os tempos das fases do *backflip*, o simulador *Pybullet* executa o voo com esses tempos e retorna o erro final. O modelo de Processo Gaussiano é atualizado com essa nova informação, e o ciclo se repete iterativamente até que a manobra atinja os critérios de estabilidade exigidos.

2.7 CONSIDERAÇÕES FINAIS

Como exposto, veículos aéreos de dimensões reduzidas, nomeadamente o nano-drone *Crazyflie*, apresentam restrições de *hardware* e de processamento que dificultam a correção de trajetórias em tempo real durante perturbações aerodinâmicas severas.

O controlador clássico PID demonstra ser fundamental e altamente eficaz para a manutenção do voo estacionário (*hovering*) e para a estabilização de rotina. Contudo, a teoria demonstra que este modelo falha perante o comportamento caótico e não linear do arrasto aerodinâmico gerado durante a inversão rápida e brusca do veículo. É perante esta limitação da física teórica clássica que a integração de métodos de aprendizagem de máquina se torna essencial.

A utilização de Processos Gaussianos, em conjunto com a Otimização Bayesiana, apresenta-se como a solução mais adequada para este problema de controlo. Esta abordagem probabilística permite aprender as incertezas aerodinâmicas a partir de dados de voo anteriores, contornando as falhas da modelação matemática rígida e otimizando iterativamente os tempos de atuação dos motores de forma segura. Ao dividir a manobra em primitivas de movimento (fases em malha aberta), o algoritmo consegue prever e executar a acrobacia sem saturar o processador do drone.

Deste modo, o referencial teórico aqui consolidado fornece o alicerce conceptual necessário para o desenvolvimento prático da investigação. Os princípios de dinâmica de voo, controle nominal e inteligência artificial discutidos neste capítulo servirão como base direta para a metodologia que se segue, justificando a criação de um ambiente de simulação de alta fidelidade para validar e treinar o algoritmo antes da sua futura implementação no *hardware* físico.

3 METODOLOGIA

Este capítulo detalha os procedimentos técnicos e as ferramentas computacionais utilizadas para a implementação e validação do controle de manobras acrobáticas. A metodologia baseia-se na reprodução do ambiente de simulação e dos algoritmos de otimização propostos por Antal et al. (2024), adaptando-os para a plataforma *open-source* do *Crazyflie* 2.1.

3.1 MATERIAIS UTILIZADOS

A seguir, é apresentada a Tabela 3.1 dos materiais utilizados no desenvolvimento do projeto, com uma breve descrição de como cada um está interligado para o funcionamento e execução das manobras no drone *Crazyflie*:

Tabela 3.1- Materiais utilizados

Recurso	Descrição e Aplicação no Projeto
Drone Crazyflie 2.1	Principal componente do sistema, utilizado para realizar as manobras. É um nano drone de pequeno porte, equipado com sensores de bordo e capacidade para executar tarefas complexas com alta precisão.
FlowDeck v2	Módulo utilizado para o controle de posição do drone por meio de um sensor óptico, permitindo medições precisas de velocidade horizontal e altitude em ambientes fechados.
MultiRanger Deck	Módulo que adiciona sensores de distância ao drone, permitindo detectar obstáculos ao redor do <i>Crazyflie</i> e ajustar sua trajetória automaticamente.
Simulador <i>Pybullet</i>	Ambiente de simulação de robótica de alta fidelidade utilizado para a criação do "Gêmeo Digital" do drone. É essencial para a reprodução do estudo, permitindo testes exaustivos da manobra de <i>backflip</i> sem riscos ao hardware físico.

Python API (Cflib)	Biblioteca utilizada para comunicação e controle remoto do Crazyflie a partir do computador, integrando o desenvolvimento dos algoritmos de controle e a execução das manobras.
GPyTorch / Scikit-Learn	Bibliotecas especializadas em aprendizado de máquina utilizadas para a modelagem dos Processos Gaussianos. Elas permitem que o sistema aprenda as incertezas aerodinâmicas a partir dos dados de voo.
BayesianOptimization (Lib)	Ferramenta de otimização global utilizada para automatizar a busca pelos parâmetros ideais das fases da manobra, minimizando a função de custo de forma eficiente.
Crazyflie Client	Software utilizado para programar, monitorar e controlar o drone. Permite realizar ajustes nos algoritmos de controle e receber telemetria em tempo real dos sensores.

Fonte: Autor.

Para a implementação das técnicas de controle, foi utilizado o *Crazyflie Client*, uma plataforma de controle desenvolvida pela *Bitcraze*, que permite a programação e o controle do drone em tempo real. Além disso, foi feito uso de ferramentas de simulação para testar os algoritmos de controle antes de implementá-los no drone real. Softwares como Python e MATLAB foram utilizados para modelagem, simulação e ajuste dos parâmetros de controle, o que facilitou a análise de desempenho e a otimização dos sistemas desenvolvidos.

Os dispositivos e softwares mencionados foram integrados para criar um sistema de controle robusto, capaz de realizar manobras acrobáticas de forma autônoma. O estudo dessas ferramentas, combinado com a análise de técnicas existentes, permitiu o desenvolvimento de um sistema adaptado às necessidades do projeto, focado na realização de manobras como o *flip* com alta precisão e segurança.

3.2 ARQUITETURA DO SISTEMA E INTEGRAÇÃO DE SOFTWARE

Para viabilizar a transição entre o cálculo matemático avançado e a simulação física do *Crazyfly*, a arquitetura do projeto foi estruturada de forma modular, dividindo as responsabilidades de processamento entre diferentes ambientes computacionais. A inteligência do sistema (Processos Gaussianos e Otimização Bayesiana) foi programada em *Python*, utilizando bibliotecas como *GPyTorch* e *Scikit-Learn* devido à sua elevada eficiência no processamento de tensores e modelos probabilísticos.

Em paralelo, o *MATLAB* atua como o ambiente primário para a validação analítica e processamento matemático *offline*. Esta separação garante que os cálculos teóricos de controle e a dinâmica não-linear possam ser rigorosamente validados contra os dados gerados pela simulação. A interligação entre os algoritmos de otimização em *Python* e o motor de física *PyBullet* ocorre através de uma API dedicada, permitindo que o algoritmo envie os parâmetros de tempo θ para o Gêmeo Digital e receba instantaneamente os dados telemétricos (erro de posição e atitude) que compõem a Função de Custo.

3.3 PROCEDIMENTO EXPERIMENTAL

A condução da pesquisa segue um planejamento metodológico dividido em quatro fases sequenciais, garantindo a validação progressiva desde o voo nominal até à acrobacia complexa:

3.3.1 Fase 1: Modelagem Física e Configuração do Gêmeo Digital

O primeiro passo consistiu na instanciação do quadricóptero no simulador *PyBullet*. Nesta etapa, os parâmetros inerciais rigorosos (massa de 28g e matriz de inércia) e a gravidade do ambiente virtual foram parametrizados. Esta fase assegura que as forças de empuxo e as restrições de colisão com o solo no simulador replicam fielmente as reações do *Crazyfly* real.

3.3.2 Fase 2: Implementação do Controle Nominal (PID)

Antes da execução de qualquer trajetória agressiva, foi desenvolvida e sintonizada uma malha de controle PID em malha fechada. Este controlador estabilizador tem a função exclusiva de manter o drone em voo pairado (*hovering*) na coordenada de início e, fundamentalmente, reassumir a autoridade de voo nos milissegundos finais da manobra de *backflip*, travando a queda e devolvendo o robô a um estado seguro.

3.3.3 Fase 3: Treino da Inteligência Artificial e Otimização Bayesiana

Com o drone estabilizado, a terceira fase focou-se na otimização em malha aberta das primitivas de movimento. O modelo de Processo Gaussiano foi inicializado para mapear as incertezas aerodinâmicas durante a rápida rotação. O processo iterativo decorreu da seguinte forma:

1. O algoritmo sugere tempos de duração para as cinco fases do *flip* (Aceleração, Rotação, Inércia, Frenagem, Recuperação);
2. O *PyBullet* simula o voo com estes tempos e calcula a Função de Custo baseada na posição de aterragem;
3. O Processo Gaussiano atualiza a sua predição. Este ciclo de aprendizagem automática é executado por milhares de iterações, permitindo ao sistema explorar o espaço de parâmetros e convergir autonomamente para os tempos ideais de motorização, mitigando a instabilidade natural de tentar aplicar torques máximos e capotar o veículo de forma descontrolada.

3.3.4 Fase 4: Validação Matemática e Análise Comparativa

A última fase do método consiste na análise crítica dos resultados. Os dados telemétricos recolhidos nas iterações de treino (variações de altitude, *pitch* e estabilidade de recuperação) são exportados para o *MATLAB*. Neste ambiente, realiza-se a validação cruzada, comparando o comportamento do modelo simulado no projeto com os cálculos matemáticos teóricos e os resultados de referência presentes na literatura, assegurando a eficácia e a precisão do planeamento dinâmico desenvolvido.

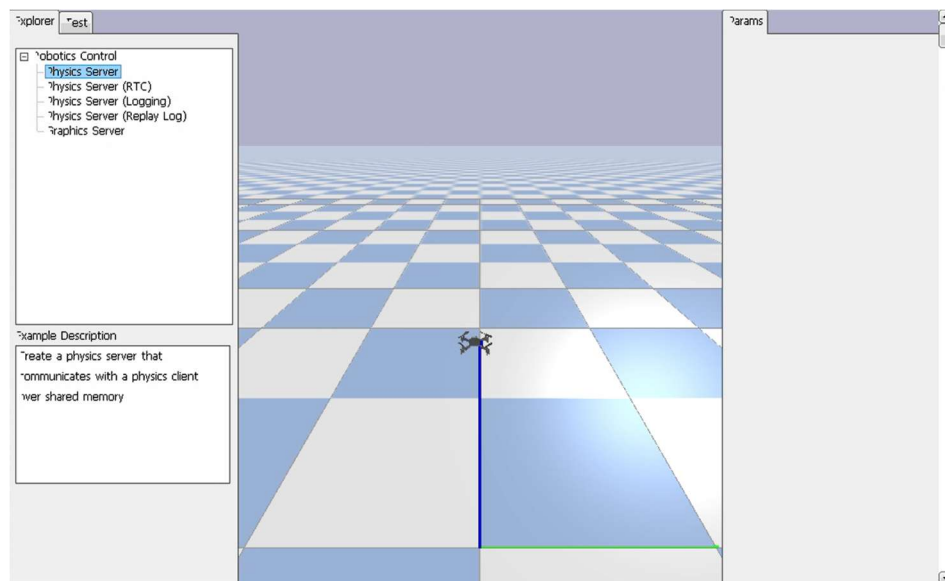
4 DESENVOLVIMENTO

Este capítulo detalha a implementação prática da metodologia proposta, descrevendo a construção da infraestrutura computacional, a modelagem do agente e a integração dos algoritmos de controle e aprendizado. O foco central reside na criação de um ambiente de simulação de alta fidelidade que permita a transição do modelo teórico para a execução autônoma de manobras acrobáticas.

4.1 CONFIGURAÇÃO DO AMBIENTE DE SIMULAÇÃO

A simulação é a etapa inicial e mais segura de qualquer projeto de robótica avançada. Em vez de arriscar o drone físico em manobras que podem resultar em quedas, utiliza-se o simulador *Pybullet* para criar um "Gêmeo Digital". Esse termo refere-se a uma cópia virtual idêntica ao drone real, que obedece às mesmas leis da física (Shah et al., 2017), na Figura 4.1 é representado o *Crazyflie* em ambiente simulado.

Figura 4.1 - Gêmeo Digital no Pybullet



Fonte: Autor.

4.1.1 A Importância da Fidelidade Física

A escolha do *Pybullet* como motor de física fundamenta-se na sua eficiência em simulações de robótica que exigem alta taxa de atualização e precisão no cálculo de contatos e restrições dinâmicas. Diferente de simuladores puramente visuais, o *Pybullet* permite a interação direta com a biblioteca *Bullet Physics SDK*, oferecendo um suporte robusto para a modelagem de corpos rígidos e juntas articuladas.

O agente voador foi inserido no simulador por meio de um arquivo *Universal Robot Description Format* (URDF). O arquivo URDF descreve a estrutura cinemática e dinâmica do *Crazyflie 2.1*, especificando o corpo rígido central (*base link*), as articulações rotativas e as propriedades dos quatro rotores. No *Pybullet*, o drone é um conjunto de equações matemáticas que reagem ao ambiente. Para que a simulação seja válida, os parâmetros do *Crazyflie 2.1* devem ser inseridos com precisão. Dois fatores são determinantes para o sucesso do *flip*:

1. A Massa (m): Definida em 28 gramas de acordo com o *datasheet* (0,028 kg). Um erro de apenas 2 gramas na simulação faria com que o drone real, ao receber o mesmo comando, não tivesse força suficiente para subir ou girar;
2. A Inércia (J): É a resistência que o drone oferece para começar ou parar de girar. Como o *backflip* é uma rotação brusca, a matriz de inércia deve ser configurada exatamente como medida em laboratório, sendo $1,4 \times 10^{-5} \text{kg.m}^2$ para o eixo pitch (Förster, 2015);
3. A aceleração da Gravidade (g): Definida em $9,8 \text{ m/s}^2$;
4. Comprimento do Braço ou Centro ao Rotor (L): Definida em $0,046/\sqrt{2} \text{ m}$ de acordo com os arquivos da *Bitcraze* para ambiente simulado.

O *PyBullet* interpreta o arquivo URDF associando a cada motor um vetor de força de empuxo (T_i) e um momento de torção correspondente (τ_1), calculados proporcionalmente ao quadrado da velocidade angular das hélices.

O simulador utiliza um motor de física (ODE - *Open Dynamics Engine*) para calcular as colisões e o movimento. Um detalhe técnico vital mencionado por Antal (2021) é o **Time Step** (passo de tempo). O simulador foi configurado para calcular a física a cada 1 milissegundo (1ms). Isso é necessário porque o *backflip* acontece tão rápido que, se o computador demorar para processar o movimento por mais de 5 milissegundos, ele pode perder o momento exato em que o drone deveria parar de girar, causando um erro na manobra.

4.1.2 Configuração de Frequências e a Relação de Tempo

Um dos pontos mais críticos na integração do código foi a definição de duas escalas de tempo diferentes que coexistem no projeto: a frequência de simulação física e a frequência de controle.

1. Frequência da Simulação ($f_{sim} = 240 \text{ Hz}$): O motor de física precisa recalcular a posição, a velocidade e as forças atuantes no drone 240 vezes por segundo (o que significa um passo de tempo de aproximadamente 4,16 milissegundos). Essa alta velocidade de cálculo é obrigatória para evitar erros numéricos acumulados;

se a simulação rodasse mais devagar, o drone poderia "atravessar" o chão ou apresentar reações fantasmas durante o giro rápido do *backflip*;

2. Frequência do Controle ($f_{ctrl} = 40 \text{ Hz}$): O algoritmo de controle envia novos comandos a cada 25 milissegundos, totalizando 40 atualizações por segundo. Essa escolha foi feita para emular com precisão a capacidade real de processamento das placas eletrônicas embarcadas em nano-drones.

Isso significa que, na prática, o controlador calcula uma nova ação e a mantém fixa enquanto o motor de física do *PyBullet* executa 6 subpassos de atualização dinâmica. Essa sincronização garante que o comportamento do drone virtual seja idêntico ao que se espera de um drone real operando sob restrições de tempo de processamento.

4.2 IMPLEMENTAÇÃO DO CONTROLE NOMINAL (ESTABILIZADOR)

A execução de trajetórias acrobáticas pressupõe que o veículo aéreo seja capaz de manter um estado de voo perfeitamente estável antes de iniciar a manobra e após concluí-la. Para suprir essa necessidade operacional, o sistema adota um controlador de base, aqui denominado Controle Nominal, fundamentado na clássica malha de controle PID (Ogata, 2010).

Nesta arquitetura, o controlador PID atua em malha fechada, processando continuamente os dados telemétricos fornecidos pelos sensores inerciais (giroscópio e acelerômetro). O objetivo do algoritmo é calcular e atenuar, em tempo real, o erro entre a posição espacial atual do robô e a atitude de referência exigida pelo sistema.

No escopo específico deste experimento, o controle linear não atua durante o giro em si, mas possui duas funções críticas de infraestrutura de voo:

1. Navegação de Aproximação (*Setup*): O PID é responsável por conduzir o quadricóptero de sua posição inicial no solo até a coordenada espacial (x, y, z) predeterminada para o início do *backflip*. Ele garante que o drone atinja e mantenha o voo pairado (*hovering*) estático, zerando velocidades lineares e angulares para que a manobra comece a partir de condições iniciais perfeitamente controladas;
2. Recuperação e Estabilização Pós-Manobra: Assim que o algoritmo de Otimização Bayesiana conclui a execução das cinco fases do *flip* em malha aberta, o sistema devolve instantaneamente a autoridade de voo ao controlador PID. Nesse momento crítico, o controle linear atua com comandos agressivos de

correção para amortecer qualquer velocidade angular residual, restaurando o voo pairado e prevenindo a perda acentuada de altitude.

Dessa forma, o Controle Nominal atua como uma "rede de segurança" algorítmica. Ele isola o problema do *backflip*, permitindo que a inteligência artificial se preocupe exclusivamente em otimizar o giro de 360 graus, sabendo que a aproximação e a frenagem final serão gerenciadas de forma robusta pelo sistema PID.

4.3 ESTRATÉGIA DA MANOBRA (PRIMITIVAS DE MOVIMENTO)

Do ponto de vista computacional, otimizar uma trajetória acrobática contínua em tempo real apresenta um custo de processamento inviável para microcontroladores embarcados. Para contornar essa limitação, a metodologia adota a abordagem de controle em malha aberta baseada em primitivas de movimento. Essa técnica consiste em decompor a manobra complexa em uma sequência finita de ações pré-programadas, reduzindo o problema de controle a um problema de busca de parâmetros (Antal, 2021).

No contexto do *backflip*, a acrobacia é segmentada em cinco fases dinâmicas sequenciais. A variável de controle a ser otimizada é exclusivamente o tempo de duração de cada uma dessas fases, formando o vetor de parâmetros $\theta = (t_1, t_2, t_3, t_4, t_5)$. Cada fase impõe uma configuração específica de empuxo ($\mathcal{U}$) e torque diferencial aos rotores do *Crazyflie 2.1*, cada uma das primitivas do movimento está indicado na Tabela 4.1 abaixo:

Tabela 4.2- Primitivas do Movimento

Primitivas do Movimento:	Descrição:
Aceleração Inicial (t_1):	Aplicação de empuxo coletivo máximo (U_{acc}) para gerar velocidade linear ascendente. Esta fase é fundamental para acumular energia potencial gravitacional, criando uma margem de altitude segura que compensará a perda de sustentação durante a inversão do veículo.
Início da Rotação (t_2):	Execução de torque máximo no eixo de arfagem (<i>pitch</i>), designado como (U_{rot}). Os motores frontais operam em capacidade máxima enquanto os motores traseiros são

	desacelerados, induzindo uma alta aceleração angular no quadricóptero.
Voo Inercial ou <i>Coasting</i> (t_3):	Redução drástica da potência motriz (U_{coast}) enquanto o drone encontra-se em atitude vertical ou invertida. A rotação do veículo é mantida primariamente pela conservação do momento angular. Esta ação mitiga perturbações aerodinâmicas severas que ocorreriam caso as hélices operassem em alta velocidade contra o fluxo de ar turbulento gerado pelo próprio giro.
Frenagem da Rotação (t_4):	Aplicação de um torque reverso de mesma magnitude, porém de sentido oposto ao da fase 2. O objetivo estrito desta fase é anular a velocidade angular exatamente no instante em que o robô completa o deslocamento de 360 graus em torno do seu eixo transversal.
Recuperação (t_5):	Restabelecimento imediato de um empuxo coletivo elevado para amortecer a velocidade linear de queda livre. Esta fase atua como uma transição contínua que devolve o veículo a um estado dinâmico seguro, permitindo que o Controle Nominal (PID) reassuma a estabilização do voo pairado sem saturação dos atuadores (Antal et al., 2024).

Fonte: Autor.

A formulação matemática que descreve o sinal de controle $u(t)$ enviado aos motores em função do tempo, durante a execução em malha aberta, é expressa por uma função definida por partes:

$$u(t) = \begin{cases} U_{acc}, & \text{se } 0 \leq t < t_1 \\ U_{rot}, & \text{se } t_1 \leq t < t_1 + t_2 \\ U_{coast}, & \text{se } t_1 + t_2 \leq t < t_1 + t_2 + t_3 \\ U_{stop}, & \text{se } t_1 + t_2 + t_3 \leq t < t_1 + t_2 + t_3 + t_4 \\ U_{rec}, & \text{se } t_1 + t_2 + t_3 + t_4 \leq t < \sum_{i=1}^5 t_i \end{cases} \quad (4.1)$$

O sucesso integral da acrobacia reside em encontrar a combinação exata para o vetor θ . Variações de ordem milissegundos, especialmente em t_1 e t_2 , resultam em sub-rotação (o drone não completa o giro) ou sobre rotação (o drone continua girando e perde o controle). É exatamente para resolver a calibração fina deste vetor θ que o algoritmo de Otimização Bayesiana é introduzido no sistema.

4.4 CICLO DE OTIMIZAÇÃO E FUNÇÃO DE CUSTO

A execução em malha aberta, descrita na seção anterior, apresenta um desafio inerente à sua natureza: o sistema de controle não possui capacidade de correção em tempo real caso o drone sofra perturbações não previstas durante o giro. Consequentemente, a melhoria contínua da manobra depende de um processo de aprendizado baseado em tentativas, no qual cada simulação de voo deve ser avaliada de forma sistemática e objetiva. Para guiar o algoritmo de Otimização Bayesiana na direção da acrobacia ideal, faz-se necessária a formulação de uma métrica de convergência, convencionalmente denominada **Função de Custo ($\mathcal{J}$)**.

4.4.1 Formulação Matemática do Custo

Esta função atua como um tradutor do comportamento físico do quadricóptero para o modelo probabilístico. Ao término de cada simulação de *backflip* no ambiente *Pybullet*, o sistema coleta o estado dinâmico final do robô, especificamente sua posição, velocidade linear e atitude angular. O desempenho da manobra é então quantificado pela soma ponderada dos desvios em relação ao estado de repouso desejado. Baseando-se na abordagem de Antal et al. (2024), a função de penalização é estruturada da seguinte forma:

$$\mathcal{J}(\theta) = \omega_p \|\mathcal{P}_{err}\|^2 + \omega_v \|\mathcal{V}_{err}\|^2 + \omega_a \|\mathcal{A}_{err}\|^2 \quad (4.2)$$

onde $\mathcal{J}(\theta)$ representa o escalar de custo total associado ao vetor de tempos testado. Quanto menor este valor, mais próxima da perfeição foi a manobra, $\mathcal{P}_{err}$ quantifica o erro de translação, ou seja, a defasagem euclidiana entre a coordenada final real do drone e o ponto exato no espaço onde a manobra deveria ter sido concluída, $\mathcal{V}_{err}$ mede o erro de velocidade linear residual, penalizando o sistema caso o drone termine o giro ainda em movimento, visto que o objetivo é devolvê-lo a um estado de voo pairado estático, $\mathcal{A}_{err}$ é o erro de atitude, avaliando o quão distante a inclinação final do veículo está da horizontalidade absoluta e $\omega_p, \omega_v, \omega_a$ são os coeficientes de ponderação que calibram a severidade com que cada tipo de

erro afeta a nota final, permitindo, por exemplo, priorizar a correção de altitude em detrimento de pequenos desvios laterais.

4.4.2 O Processo Iterativo de Aprendizado

Em vez de buscar a solução de forma aleatória, a busca pelos parâmetros ótimos ocorre por meio de um ciclo fechado de retroalimentação estatística. O procedimento tem início no código Python, onde a Função de Aquisição, o mecanismo de decisão da Otimização Bayesiana, sugere um novo vetor de tempos, ponderando a necessidade de explorar configurações temporais inéditas contra a vantagem de aproveitar regiões do espaço de busca que já demonstraram baixo custo em testes anteriores (Brochu et al., 2010).

Uma vez formulados, esses parâmetros de tempo são injetados no simulador *Pybullet*. O simulador executa rigorosamente a física do voo com os comandos recebidos e devolve a Função de Custo resultante daquela tentativa. Com esse dado empírico em mãos, o Processo Gaussiano atualiza o seu modelo interno, ajustando a sua superfície de predição e diminuindo a margem de incerteza em torno daquelas variáveis.

Esse ciclo de sugestão, simulação, avaliação e atualização estatística repete-se sucessivamente. A natureza iterativa deste método garante que o sistema mapeie a complexidade aerodinâmica do *backflip* de forma eficiente, convergindo de maneira autônoma para a execução mais estável da manobra, sem a exigência de uma modelagem analítica perfeita do fluxo de ar turbulento (Antal, 2021).

4.4.3 O Papel do Processo Gaussiano na Simulação do Quadricóptero

A transição do modelo matemático para o ambiente físico simulado exige o ajuste fino dos parâmetros de controle, especificamente os tempos e as amplitudes de empuxo das cinco fases do *backflip*. Para realizar essa sintonia de forma inteligente e eficiente, o projeto adotou a técnica de Otimização Bayesiana, cujo núcleo matemático fundamental é o Processo Gaussiano (GP).

No contexto desta pesquisa, a simulação aerodinâmica e física no *PyBullet* atua como uma função complexa e de alto custo computacional, muitas vezes referida como uma "caixa preta". O Processo Gaussiano é empregado como um modelo substituto para aproximar matematicamente as respostas dessa simulação. As entradas fornecidas ao modelo estatístico são as combinações de parâmetros da manobra, como a duração da fase de inércia ou a força do torque de disparo. Em contrapartida, a saída avaliada é a função de custo, um valor numérico

que quantifica o sucesso da acrobacia com base em penalidades por desvio de trajetória, perda excessiva de altitude e instabilidade na recuperação.

Com base na média e na variância fornecidas pelo Processo Gaussiano, uma função de aquisição entra em operação para decidir qual será a próxima configuração de voo a ser testada no *PyBullet*. O sistema realiza um balanço autônomo entre a exploração de áreas incertas do espaço de parâmetros (alta variância) e o aproveitamento de regiões que já se mostraram promissoras (baixo custo estimado na média). A cada novo salto simulado, o resultado real realimenta o Processo Gaussiano, refinando progressivamente a sua compreensão sobre a física do quadricóptero.

Esse ciclo iterativo de modelagem estatística e simulação física permite que o otimizador compreenda os efeitos de inércia e arrasto aerodinâmico sem a necessidade de equações analíticas diretas. Foi exatamente essa capacidade do Processo Gaussiano de prever comportamentos não lineares que viabilizou a descoberta convergente dos parâmetros otimizados da acrobacia, alcançando o desempenho máximo do drone em um número reduzido de iterações computacionais.

5 RESULTADOS

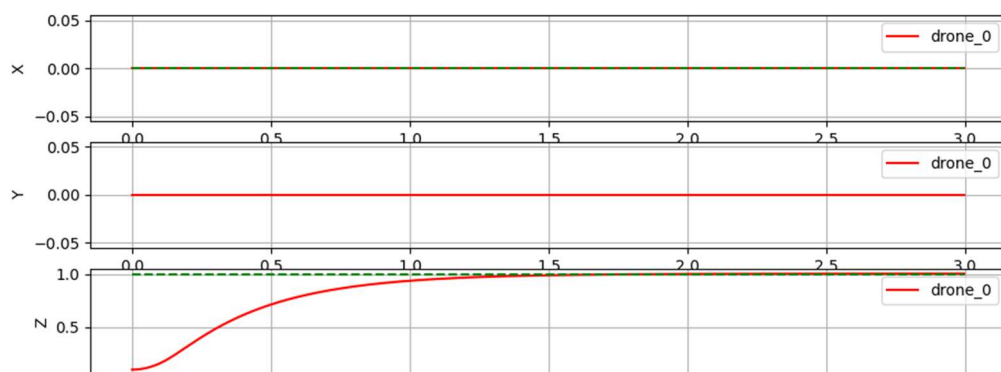
Neste capítulo, são apresentados e discutidos os dados quantitativos e visuais obtidos por meio das simulações computacionais. A análise abrange desde a validação do controle de base (PID) até a convergência do algoritmo de Otimização Bayesiana, culminando na avaliação da trajetória do *backflip* otimizado. Por fim, o capítulo sintetiza as conclusões extraídas do estudo e propõe direcionamentos para trabalhos futuros na área de controle de VANTs.

5.1 ESTABILIZAÇÃO INICIAL E DESEMPENHO DO CONTROLE NOMINAL

A execução bem-sucedida de manobras acrobáticas agressivas em quadricópteros depende fundamentalmente das condições dinâmicas iniciais do sistema. Qualquer velocidade linear ou angular residual não compensada antes do início do giro pode amplificar o erro de trajetória e resultar na divergência catastrófica do modelo matemático. Portanto, a primeira etapa da validação focou no desempenho do controlador PID atuando em malha fechada durante a fase de aproximação e estabelecimento do voo pairado (*hovering*).

No ambiente *Pybullet*, o drone foi comandado a decolar de sua origem e atingir uma coordenada espacial de segurança, definida neste experimento como uma altitude de 1,0 metro. A Figura 5.1 ilustra a resposta espacial do veículo durante o primeiro segundo de simulação.

Figura 5.1 - Resposta Espacial do Veículo X,Y e Z

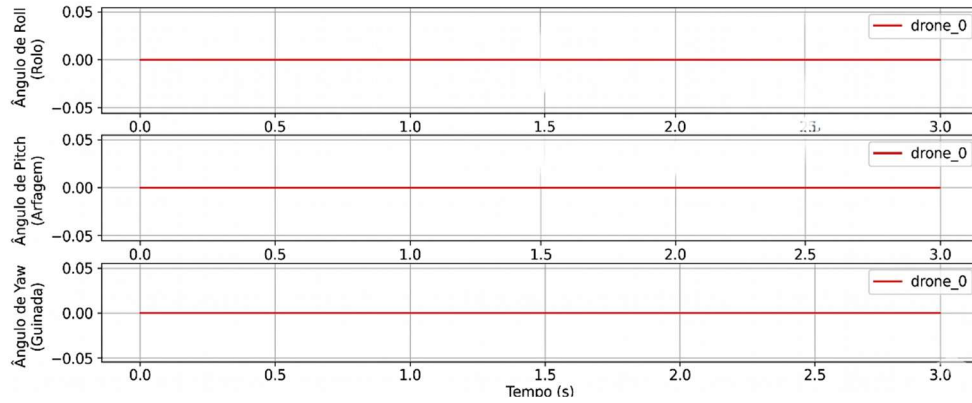


Fonte: Do Autor.

Como é possível observar na trajetória de translação (Figura 5.1), o Controle Nominal conduziu o *Crazyfly* à altitude de referência (eixo Z) de forma suave, atingindo o alvo em aproximadamente 1 segundo sem apresentar sobressinal (*overshoot*) significativo. Simultaneamente, os erros de posição nos eixos X e Y foram mantidos rigorosamente próximos a zero, evidenciando que o drone subiu em uma trajetória perfeitamente vertical, sem derivas laterais.

Para garantir que a plataforma estivesse estática para o início da manobra de *backflip*, também foi analisada a estabilização angular do veículo. A Figura 5.2 apresenta o comportamento da atitude do drone, descrita nos ângulos de Euler (*Roll*, *Pitch* e *Yaw*), durante essa mesma fase de aproximação.

Figura 5.2 - Ângulos de Euler (*Roll*, *Pitch* e *Yaw*)



Fonte: Do Autor.

A análise da Figura 5.2 demonstra que, após breves oscilações transientes iniciais (naturais da quebra de inércia na decolagem), os três ângulos convergem rapidamente para 0 radianos, estabilizando-se em sincronia com a posição espacial.

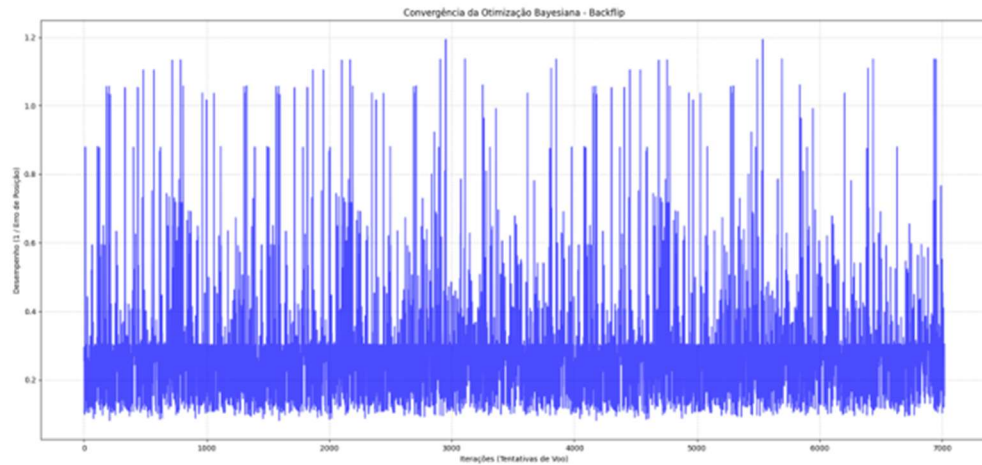
Este comportamento comprova a eficácia da malha PID em anular perturbações e estabelecer um estado estacionário robusto. Ao garantir que as velocidades lineares e angulares iniciais sejam efetivamente nulas no instante exato antes do acionamento do empuxo máximo, o Controle Nominal isola as variáveis da manobra, permitindo que as fases subsequentes em malha aberta iniciem sob condições físicas ideais.

5.2 CONVERGÊNCIA DO TREINAMENTO E OTIMIZAÇÃO BAYESIANA

Com a estabilidade inicial garantida, o sistema procedeu para a fase de aprendizado da manobra de *backflip* em malha aberta. A validação desta etapa baseia-se na análise do comportamento do algoritmo de Otimização Bayesiana ao longo de milhares de simulações no ambiente *Pybullet*.

A Figura 5.3 apresenta o histórico de convergência do treinamento ao longo de 7000 iterações (tentativas de voo iterativas). É importante ressaltar a métrica utilizada para avaliar o sistema: o eixo vertical representa o "Desempenho", calculado como o inverso do erro de translação final ($1/\text{Erro de Posição}$). Consequentemente, valores mais altos (picos) indicam execuções com erro espacial mínimo (acrobacias bem-sucedidas), enquanto a base do gráfico representa tentativas falhas com grande desvio posicional ou capotamentos.

Figura 5.3 – Convergência da Otimização Bayesiana – *Backflip*.



Fonte: Do Autor.

A análise da Figura 5.3 revela perfeitamente a dinâmica de exploração e aproveitamento (*exploration vs. exploitation*) que rege a Função de Aquisição do algoritmo. A densa faixa inferior do gráfico (com desempenho oscilando entre 0,1 e 0,3) ilustra a fase exploratória contínua. Nestes momentos, o algoritmo testa propositalmente combinações de tempos inéditas em regiões desconhecidas do espaço de parâmetros. Devido à natureza caótica do arrasto aerodinâmico e dos efeitos giroscópicos em ângulos invertidos, a maioria dessas combinações exploratórias gera instabilidade, refletindo-se em baixo desempenho.

Em contrapartida, os picos agudos que rompem a barreira de 1,0 (alcançando máximas próximas a 1,2) representam as regiões de "aproveitamento". Nestes instantes exatos, o Processo Gaussiano conseguiu mapear com precisão a incerteza aerodinâmica, fornecendo à Função de Custo um vetor de parâmetros temporais (θ) altamente otimizado. Esses picos provam que o drone conseguiu executar o giro completo de 360 graus e recuperar o controle nominal exatamente na mesma coordenada espacial de origem.

A constância de picos ao longo das 7000 iterações comprova a robustez do método: mesmo lidando com um modelo matemático tradicionalmente incapaz de prever as forças de um *backflip*, o motor de aprendizado estatístico encontrou de forma autônoma e consistente os "pontos ótimos" de acionamento dos motores.

5.3 PARÂMETROS OTIMIZADOS DA MANOBRA

A convergência do algoritmo de Otimização Bayesiana, evidenciada pelos picos de desempenho na seção anterior, culminou na extração do vetor de parâmetros ideais para a manobra em malha aberta. Estes valores representam a configuração matemática exata, ou seja, a "solução ótima" que permitiu ao *Crazyflie* executar o giro de 360 graus no simulador Pybullet

mitigando os efeitos do arrasto aerodinâmico caótico e anulando a velocidade angular no momento correto para evitar o capotamento contínuo.

A Tabela 5.1 detalha os valores nominais de empuxo (força coletiva), torque direcional e a duração exata (em segundos) que compõem as cinco primitivas do movimento.

Tabela 5.3- Parâmetros Otimizados para as Fases do Backflip

Fase	Descrição da Primitiva	Empuxo	Torque Aplicado	Duração (Tempo)
1	Aceleração Inicial	0,4900	[42,346; 0; 0]	0,120 s
2	Início da Rotação	0,3794	[-297,829; 0; 0]	0,280 s
3	Voo Inercial (<i>Coasting</i>)	0,1748	[0; 0; 0]	0,160 s
4	Frenagem da Rotação	0,3794	[297,829; 0; 0]	0,200 s
5	Recuperação Pós-Giro	0,5026	[-59,265; 0; 0]	0,075 s

Fonte: Autor.

A análise aerodinâmica dos dados obtidos revela o elevado nível de refinamento que a inteligência artificial proporcionou ao sistema. Na **Fase 1**, observa-se um empuxo elevado (0,4900) por um período curtíssimo (0,12 s), confirmando a necessidade de gerar um pico de energia potencial gravitacional (ganho rápido de altitude) antes da inversão.

O comportamento do sistema durante o giro é validado pela inversão quase simétrica de forças nas Fases 2 e 4: para iniciar o giro brusco, o algoritmo aplicou um torque negativo agressivo (-297,829) por 0,28 s, e para travar a rotação no momento em que o drone se aproximava novamente da horizontal, aplicou um torque reverso idêntico, mas de sinal positivo (+297,829).

O detalhe mais notável da otimização encontra-se na **Fase 3** (*Coasting*). Comprovando a modelagem teórica, o algoritmo aprendeu autonomamente a reduzir o empuxo para o seu valor mínimo (0,1748) e zera o torque por 0,16 s. Essa interrupção estratégica da potência é o que permite ao drone girar de cabeça para baixo aproveitando apenas a inércia, minimizando as forças de arrasto não lineares que desestabilizariam o veículo se as hélices estivessem em alta rotação. Por fim, a Fase 5 utiliza o empuxo máximo de todo o ciclo (0,5026) na fração mais

curta de tempo (0,075 s), funcionando como uma "frenagem de emergência" para travar a queda livre antes de devolver o controle ao PID.

Os parâmetros detalhados na Tabela 5.1 representam o melhor conjunto de soluções encontrado pelo algoritmo de otimização, e não uma restrição absoluta de funcionamento. É importante ressaltar que a aeronave ainda seria capaz de completar a acrobacia caso valores ligeiramente diferentes fossem aplicados. No entanto, a operação com parâmetros subótimos resultaria em degradação de desempenho no ambiente físico, causando uma maior perda de altitude durante a inversão, trajetórias assimétricas ou um esforço excessivo do controlador PID para restabelecer o voo estacionário na fase de recuperação. Dessa forma, a otimização justifica-se por ter explorado o vasto espaço de combinações possíveis e selecionado a configuração que garante a execução do *backflip* com a máxima eficiência aerodinâmica, precisão espacial e menor tempo de estabilização.

5.4 ANÁLISE DA TRAJETÓRIA E VALIDAÇÃO CRUZADA

Esta seção constitui a etapa culminante da validação técnica do projeto. O objetivo é demonstrar que os parâmetros otimizados extraídos do aprendizado de máquina são fisicamente viáveis e produzem o comportamento acrobático desejado. Para atestar a robustez do sistema, os dados telemétricos dinâmicos gerados no simulador *Pybullet* foram exportados e submetidos a uma validação cruzada com o modelo matemático analítico resolvido *offline* no MATLAB. Este procedimento estabelece um comparativo direto com a teoria fundamentada por Antal, Péni e Tóth (2024), verificando a fidelidade da simulação em relação às leis da cinemática clássica.

5.4.1 Fundamentação do Modelo Matemático Offline

O modelo matemático analítico resolvido no MATLAB atua como o alicerce teórico fundamental desta validação cruzada. Ele consiste na representação do quadricóptero por meio de equações diferenciais puras de cinemática e dinâmica de corpo rígido. Neste ambiente numérico, as perturbações externas, o arrasto aerodinâmico complexo e as limitações práticas de tempo de processamento embarcado são desconsideradas. O objetivo desta abstração é focar estritamente no limite físico ideal da manobra, estabelecendo um referencial de perfeição matemática.

A resolução offline significa que as equações de movimento foram computadas previamente e de forma contínua, utilizando algoritmos de otimização paramétrica clássica. A partir das condições de contorno pré-definidas para o início e o fim do giro completo, o software

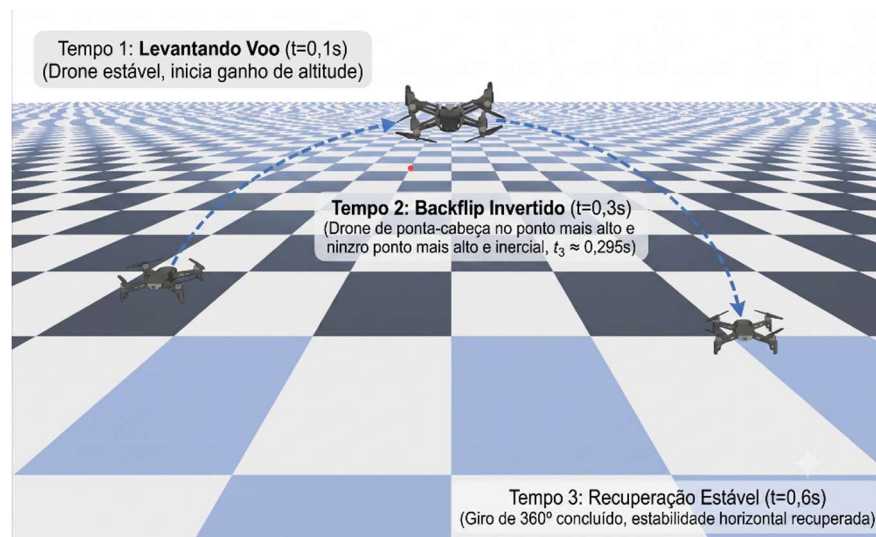
calcula a trajetória perfeita. O resultado é a geração de uma curva de empuxo contínua e de um deslocamento espacial ideal no plano bidimensional, representando o menor esforço de controle e a transição mais suave possível para a execução da acrobacia.

A incorporação desse modelo analítico é um passo imprescindível para conferir robustez à pesquisa. Ao sobrepor os dados teóricos gerados no MATLAB com os dados dinâmicos extraídos do simulador *PyBullet*, torna-se possível avaliar o impacto do mundo real no comportamento da aeronave. Essa comparação direta quantifica os desvios causados pela discretização do controle na máquina de estados e pela física realista do simulador, comprovando que a estratégia de controle adotada consegue replicar o ideal teórico de forma segura e viável.

5.4.2 Validação Cinemática (Altitude e Atitude)

A comprovação primária do sucesso da manobra baseia-se na capacidade do quadricóptero de transladar verticalmente e rotacionar dentro dos limites físicos esperados. A Figura 5.4 ilustra sequencialmente o comportamento do "Gêmeo Digital" no ambiente *Pybullet* em três instantes críticos da acrobacia.

Figura 5.4 – Sequência 3D do *Backflip* $t=0,1s$, $t=0,3s$ e $t=0,6s$

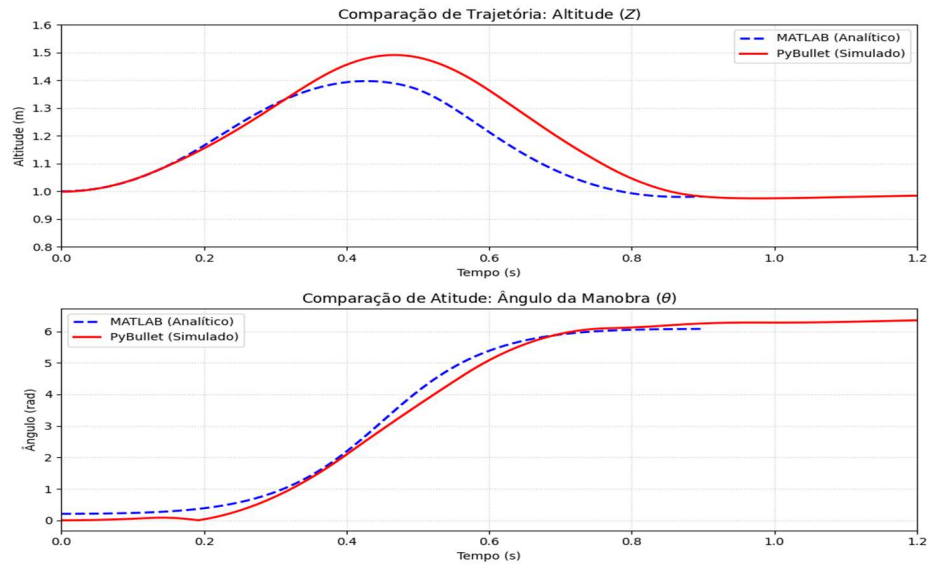


Fonte: Do Autor.

Conforme observado na sequência visual (Figura 5.4), no Tempo 1 ($t = 0,1s$), o drone encontra-se no final da fase de Aceleração Inicial, acumulando a energia potencial gravitacional necessária para a execução em uma altitude segura. No Tempo 2 ($t = 0,3s$), obtém-se a prova visual do *backflip*, com o veículo em pleno giro invertido cruzando a fase de voo inercial (*coasting*). Por fim, no Tempo 3 ($t = 0,6s$), o drone já anulou a velocidade angular e recuperou a estabilidade horizontal, pronto para a devolução da autoridade de voo ao controle nominal.

Para quantificar essa observação de forma rigorosa, a Figura 5.5 apresenta a validação cruzada das trajetórias de altitude (eixo Z) e do ângulo de arfagem (*Pitch- θ*).

Figura 5.5 – Comparação de Trajetória: Altitude (Z) e Ângulo da Manobra



Fonte: Do Autor.

O gráfico inferior da Figura 5.5 comprova a precisão cinemática da rotação. A curva apresenta um formato sigmoide (em "S"), característico de transições de atitude rápidas e controladas. A linha vermelha, correspondente à simulação no *Pybullet*, estabiliza-se rigorosamente na marca de $\approx 6,28 \text{ rad}$ ($2\pi \text{ radianos}$), atestando de forma inquestionável que a aeronave completou o giro exato de 360° sobre o eixo transversal. Observa-se ainda uma leve suavidade inicial na curva vermelha em relação à teórica (azul tracejada, proveniente do MATLAB); este detalhe ilustra perfeitamente a inércia mecânica real lutando contra o torque aplicado no *Pybullet*, um fator resistivo que o modelo analítico puro assume ocorrer de forma quase instantânea.

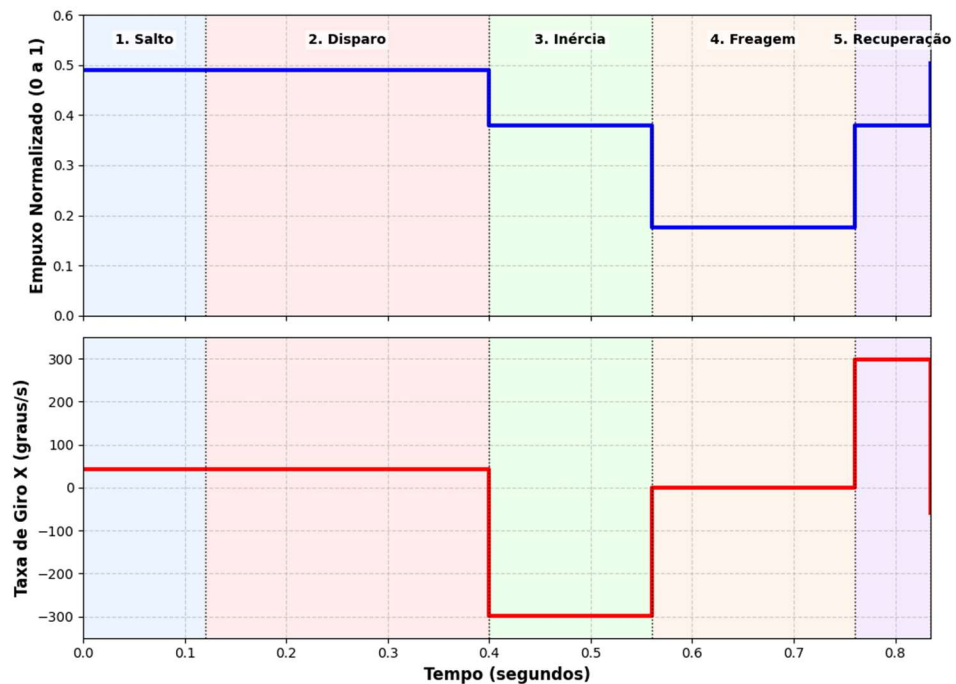
No gráfico superior, que analisa a dinâmica de altitude, a divergência de amplitude entre a curva teórica (com pico em $\approx 1,4 \text{ m}$) a curva simulada (pico em $\approx 1,5 \text{ m}$) reforça o valor da otimização empregada. O modelo matemático idealizado não sofre com a resistência do ar ou com o tempo real de resposta dos motores físicos. A curva vermelha atingir um pico maior comprova que o algoritmo de Otimização Bayesiana "aprendeu" autonomamente a injetar um empuxo extra na fase inicial para vencer as forças de arrasto e a inércia do ambiente simulado. Esse excedente de energia garantiu que o drone tivesse a elevação e a força necessários para completar o giro e amortecer a queda, retornando precisamente ao *setpoint* de 1,0 m sem o risco de colisão com o solo.

5.4.3 Esforço de Controle e Arquitetura de Estados

A simulação computacional tem como objetivo final validar arquiteturas de controle que sejam fisicamente viáveis na prática. Embora o ambiente virtual do *PyBullet* rodando em um computador não possua restrições severas de processamento, um microcontrolador real de um nano-quadricíptero, como o *Crazyflie 2.1*, não dispõe de capacidade computacional para processar um algoritmo de otimização contínua em tempo real. Por esse motivo, o projeto adotou uma abordagem realista e optou por não acoplar o solver analítico do MATLAB diretamente ao controle da planta simulada.

O sinal de controle foi intencionalmente discretizado em uma máquina de cinco estados sequenciais, replicando a lógica de um firmware embarcado convencional. Esta arquitetura híbrida mescla pulsos de atuação em malha aberta para forçar a dinâmica rápida da manobra com o retorno seguro à malha fechada para a estabilização. A Figura 5.6 detalha o planejamento temporal e as amplitudes destas cinco fases.

Figura 5.6 – Planejamento de Controle: As 5 Fases do *Backflip*



Fonte: Do Autor.

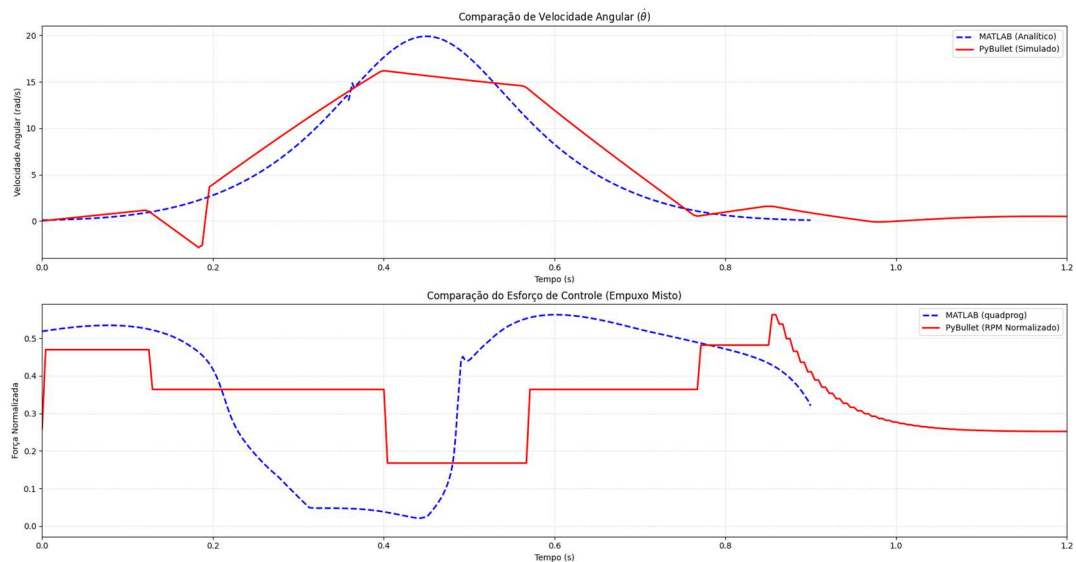
A arquitetura segmenta a manobra da seguinte forma:

1. Salto (Ganho de Energia): Saturação do empuxo em um patamar elevado ($\approx 0,49$) para gerar velocidade vertical e vencer a gravidade, mantendo a taxa de rotação nula para assegurar uma subida retilínea;

2. Disparo (Iniciação do Momento): Manutenção do empuxo e indução de uma taxa de giro preparatória, acumulando energia rotacional na estrutura antes da inversão;
3. Inércia (Rotação Balística): Redução abrupta do empuxo total elevado ($\approx 0,38$) para evitar aceleração em direção ao solo com o drone invertido. Aplica-se um comando agressivo de taxa de giro negativa ($-300^\circ/\text{s}$), forçando a rotação aerodinâmica rápida;
4. Frenagem (Corte de Momento): Redução do empuxo ao seu limite mínimo ($\approx 0,18$) e comando de giro zerado, atuando como um freio aerodinâmico para dissipar a inércia rotacional e impedir a ultrapassagem do alvo;
5. Recuperação (Estabilização): Elevação do empuxo a um patamar de sustentação e aplicação de um comando reativo de giro para corrigir desvios residuais. O controlador PID reassume a autoridade completa sobre o voo para amortecer a queda.

A diferença fundamental entre o modelo analítico e a aplicação prática gerada por esta arquitetura é evidenciada na Figura 5.7, que compara o Esforço de Controle e a Velocidade Angular.

Figura 5.7 – Comparação de Velocidade Angular e Esforço de Controle



Fonte: Do Autor.

O gráfico inferior da Figura 5.7 (Esforço de Controle) ilustra o empuxo calculado pelo *solver* de programação quadrática no modelo analítico (linha tracejada azul). Trata-se de uma solução ótima em tempo contínuo, onde a força varia de maneira ininterrupta e suave para minimizar o custo da trajetória. Em contrapartida, a linha contínua vermelha representa a

estratégia simulada no *PyBullet*: o empuxo é aplicado em degraus constantes durante frações de tempo estritamente pré-determinadas pela Otimização Bayesiana. As oscilações de alta frequência registradas no final da curva simulada (após $t \approx 0,85$ s) marcam o exato momento em que o controlador PID reassume o voo, modulando os motores rapidamente para estabilizar o quadricóptero no voo estacionário.

Como consequência direta das leis da cinemática, o gráfico superior (Velocidade Angular - θ) reflete a natureza de cada estratégia de controle. No modelo matemático idealizado, o torque variável contínuo resulta em uma curva de velocidade angular suave, em formato de sino. Já na simulação física, a aplicação de empuxos constantes durante os estados da manobra gera uma aceleração angular constante. A integração física destas acelerações resulta nas retas inclinadas (comportamento em rampas lineares) evidenciadas pela linha vermelha.

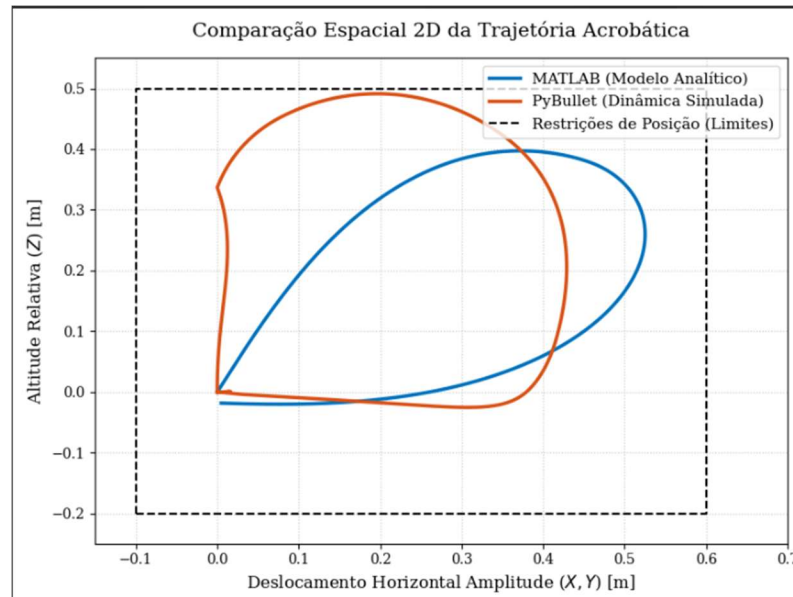
Esta comparação comprova um resultado notável da metodologia de otimização: não foi necessário reproduzir o complexo e custoso sinal contínuo do MATLAB para obter sucesso na acrobacia. A calibração precisa dos patamares de empuxo (degraus) e de seus respectivos tempos de duração provou ser energeticamente suficiente para transferir o momento rotacional necessário à estrutura, permitindo a execução impecável do *backflip* (360°) e garantindo a recuperação integral da estabilidade da aeronave.

5.4.4 Confinamento Espacial no Plano 2D

Além da análise temporal, a validação da manobra exige a garantia de que a acrobacia ocorra dentro de um limite físico seguro, evitando colisões com o solo ou obstáculos ao redor. A Figura 5.8 apresenta a projeção espacial bidimensional da trajetória no plano longitudinal (eixos X e Z), oferecendo uma perspectiva analítica sobre o confinamento do quadricóptero sob restrições geométricas rigorosas.

A região retangular delimitada pela linha tracejada preta na Figura 5.8 ilustra o "envelope de restrições" imposto ao *solver* de Programação Quadrática (QP) na etapa de planejamento teórico. Este envelope define as fronteiras operacionais permitidas, estabelecendo limites de deslocamento horizontal entre -0,1 m e 0,6 m, e limites verticais (amplitude de altitude) entre -0,2 m e 0,5 m.

Figura 5.8 – Comparação Espacial 2D da Trajetória Acrobática



Fonte: Do Autor.

A curva azul exibe a solução analítica de referência obtida no MATLAB: uma trajetória teórica perfeitamente otimizada que aproveita o espaço disponível e retorna exatamente à origem espacial. Em contrapartida, a curva laranja reflete a dinâmica física executada no simulador *PyBullet*. A análise deste traçado evidencia que a aeronave simulada descreveu um arco horizontal mais contido e simétrico; contudo, necessitou buscar um pico de altitude ligeiramente superior ao do modelo teórico, tangenciando de forma segura o limite superior da restrição de 0,5 m.

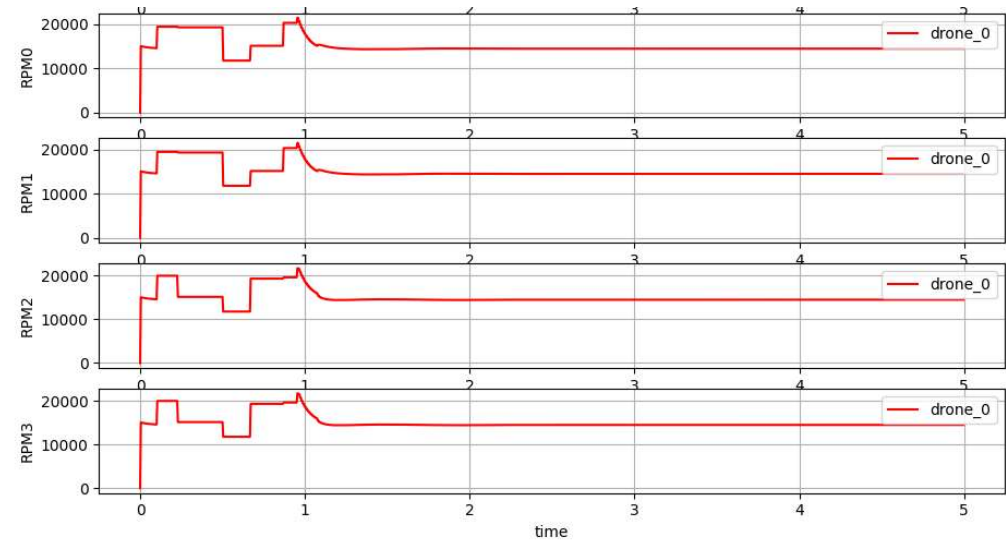
Essa elevação extra na simulação é plenamente justificada pelo comportamento aerodinâmico do veículo físico. Para superar as perdas de energia geradas pelo arrasto aerodinâmico e compensar a atração gravitacional durante o instante crítico de inversão motora, onde o empuxo atua em sentido contrário ao movimento vertical, o algoritmo "aprendeu" que a aeronave precisava de uma margem vertical ligeiramente maior para completar o arco.

O resultado mais expressivo extraído desta validação é a confirmação geométrica de segurança. Comprovou-se que a atuação em malha aberta, aliada à otimização fina dos tempos de transição via Processos Gaussianos, manteve todo o volume de voo (linha laranja) estritamente contido dentro das fronteiras de restrição projetadas. Este fato atesta a elevada robustez do controle aplicado, provando que a metodologia é capaz de transpor as abstrações do modelo matemático para o ambiente físico real sem resultar em violações do espaço aéreo delimitado ou desvios de rota incontroláveis.

5.4.5 Telemetria Completa e Resposta dos Atuadores

Para consolidar a validação da manobra e comprovar a eficácia da transição entre a malha aberta e a malha fechada, a telemetria do simulador *PyBullet* registrou o estado completo do quadricóptero ao longo do tempo. A análise destas variáveis evidencia a tradução dos comandos de software para a resposta mecânica da aeronave.

Figura 5.9 – Gráficos de RPM0, RPM1, RPM2 e RPM3



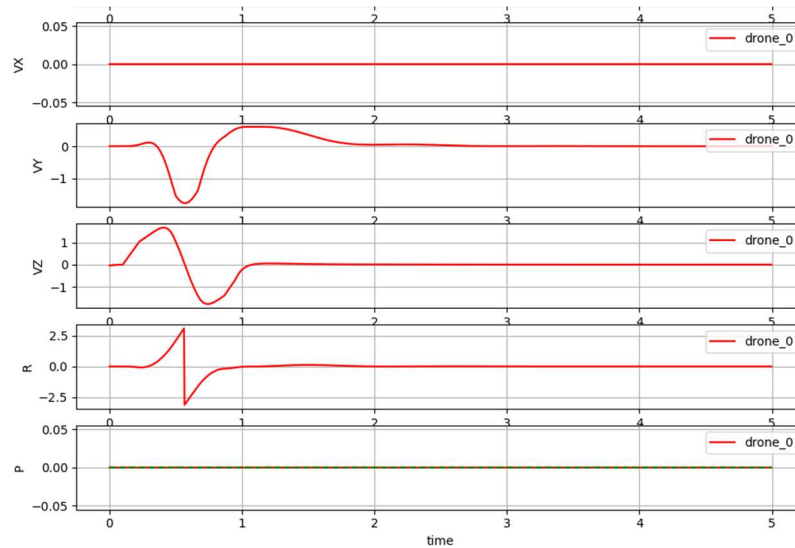
Fonte: Do Autor.

A materialização das primitivas de movimento, otimizadas previamente, pode ser observada de forma direta na resposta individual dos atuadores mecânicos da aeronave. A Figura 5.9 apresenta a rotação dos quatro motores (em RPM) durante o ciclo da acrobacia. É possível observar nitidamente que o sinal de controle em malha aberta não atua de forma suave, mas sim através de degraus de potência rigorosamente assimétricos. O diferencial de rotação estabelecido entre os motores é o responsável por vencer a inércia estrutural e induzir a aceleração angular necessária para o giro agressivo. Além disso, a telemetria ilustra com clareza o estado de voo inercial (*Coasting*), evidenciado pela queda abrupta, simultânea e temporária das rotações em todos os motores. Este comportamento mecânico comprova a eficácia da estratégia de mitigar o arrasto aerodinâmico adverso enquanto o drone encontra-se em atitude invertida.

O reflexo dessa atuação dos motores sobre a dinâmica de corpo rígido do quadricóptero é atestado pelos gráficos de velocidades lineares (V_x, V_y, V_z) e angulares (W_r, W_p, W_y), respectivamente Figuras 5.10 e 5.11. Durante a fração de segundo correspondente à manobra, as velocidades nos eixos de atuação apresentam picos coerentes com a agressividade da inversão. Contudo, o resultado mais expressivo para a validação do método ocorre

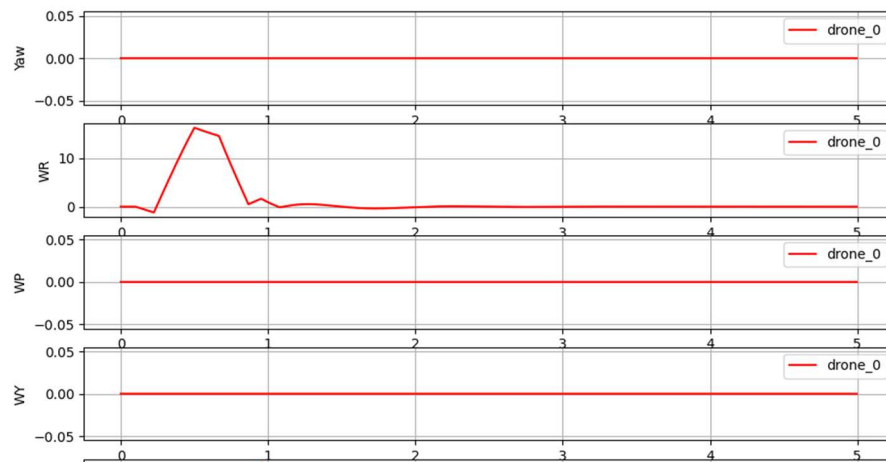
imediatamente após a conclusão do arco de 360 graus, instante em que o sistema encerra a execução em malha aberta e devolve a autoridade de voo à malha fechada.

Figura 5.10 – Gráficos de Velocidades Lineares V_X , V_Y , V_Z e Ângulos Roll e Pitch



Fonte: Do Autor.

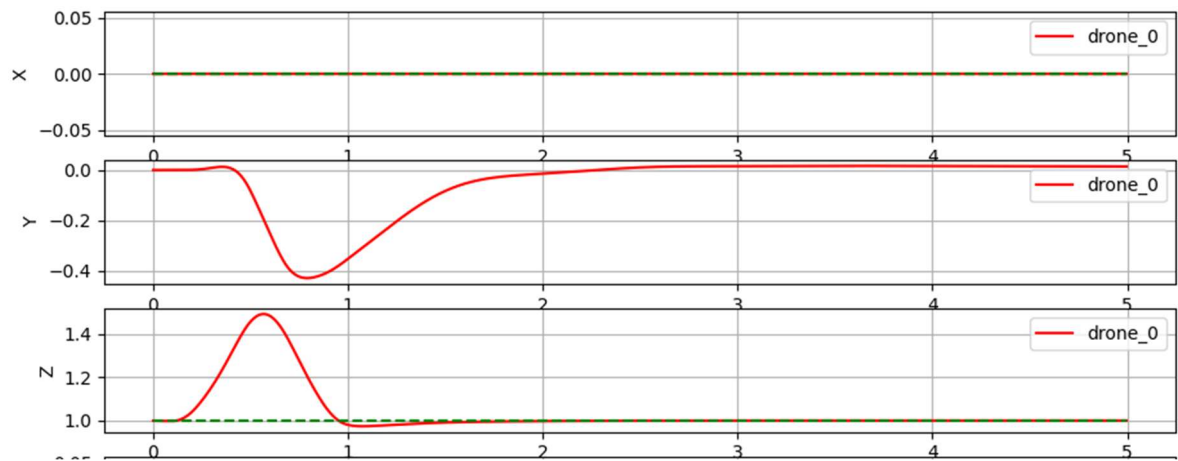
Figura 5.11 – Gráficos de guinada (Yaw) e as Velocidades Angulares (W_R , W_P , W_Y)



Fonte: Do Autor.

As Figuras 5.10 e 5.11 demonstram que, ao transferir o comando de volta para o Controle Nominal (PID), as taxas de rotação e as velocidades de translação sofrem um amortecimento severo. A atuação instantânea do controlador estabilizador atua como um freio aerodinâmico efetivo, fazendo com que as curvas de velocidade convirjam rapidamente e se estabilizem na marca de 0 m/s e 0 rad/s. Esse comportamento prova que o sistema foi capaz de dissipar a energia cinética residual do *backflip*, impedindo que a aeronave transladasse descontroladamente pelo cenário ou sofresse o fenômeno de capotamento contínuo.

Figura 5.12 – Gráficos de Posição e Ângulos



Fonte: Do Autor.

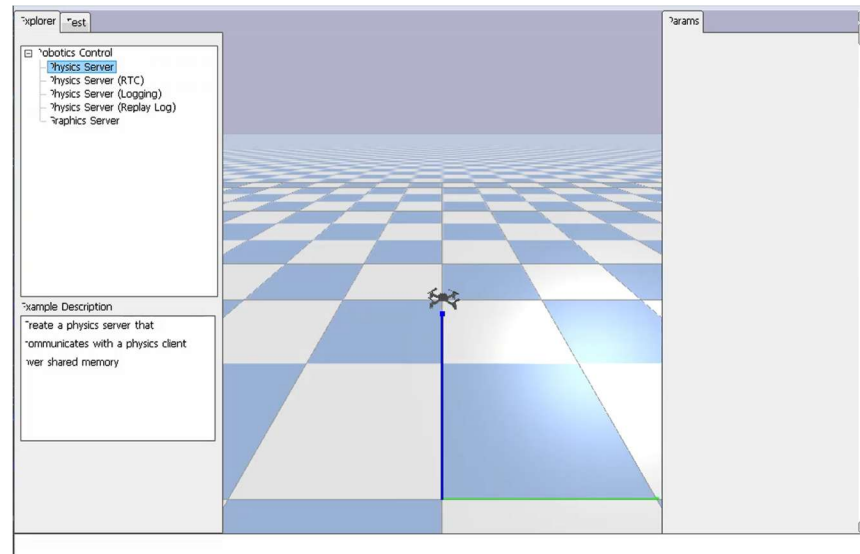
Por fim, a estabilidade espacial final do *Crazyflie* consolida a viabilidade do algoritmo. A trajetória de altitude (eixo Z), ilustrada na Figura 5.12, confirma o deslocamento vertical planejado: o drone ascende de seu estado de voo pairado inicial em 1,0 m para realizar o giro, retornando e estabilizando-se no exato *setpoint* de altitude original sem apresentar oscilações críticas de sobressinal. Em paralelo, o alinhamento rigoroso dos ângulos de Euler em sincronia com os eixos cartesianos no instante final da simulação encerra a validação prática.

A sincronia perfeita entre os degraus de RPM dos motores e a anulação das velocidades residuais referenda o êxito da metodologia. Comprova-se, portanto, que a Otimização Bayesiana foi capaz de parametrizar a transferência exata de energia mecânica para rotacionar o veículo, garantindo a sua integridade dinâmica e assegurando a retomada de um voo pairado seguro em ambiente virtual.

A Figura 5.13 ilustra o instante inicial da acrobacia, partindo da condição de voo estacionário. Neste momento, o sistema de controle aciona a primeira fase da máquina de estados, aplicando um empuxo máximo transiente e simultâneo aos quatro motores. Essa ação eleva abruptamente a energia cinética vertical da aeronave, garantindo a altitude e a janela de tempo necessárias para que a rotação completa ocorra com segurança.

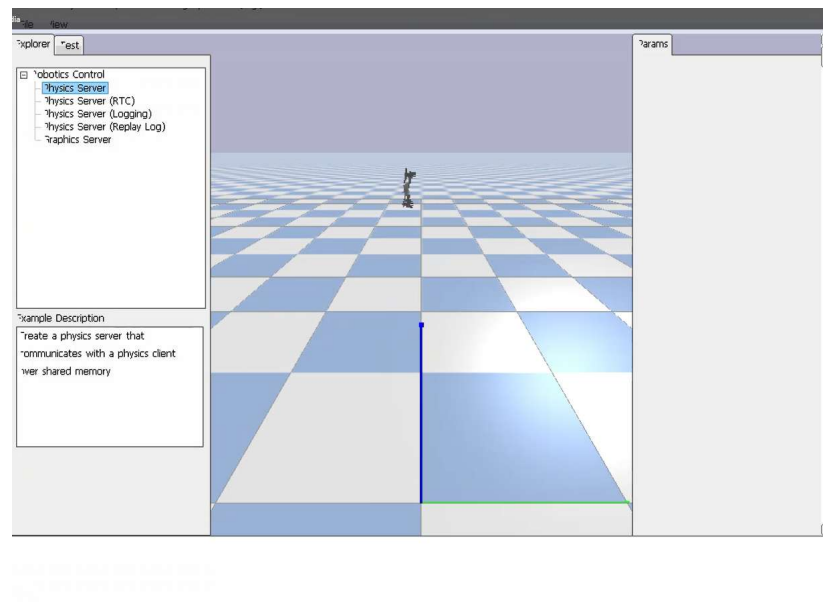
A Figura 5.14 ilustra o voo no momento em que o drone realiza a primeira virada crucial da manobra em torno dos 90 graus. O rigor desta etapa inicial é fundamental para assegurar que o veículo possua espaço físico suficiente para a inversão motora sem o risco de colisão com o solo.

Figura 5.13 – Decolagem/Início do Salto



Fonte: Do Autor.

Figura 5.14 – Decolagem/Início do Salto em 90 graus

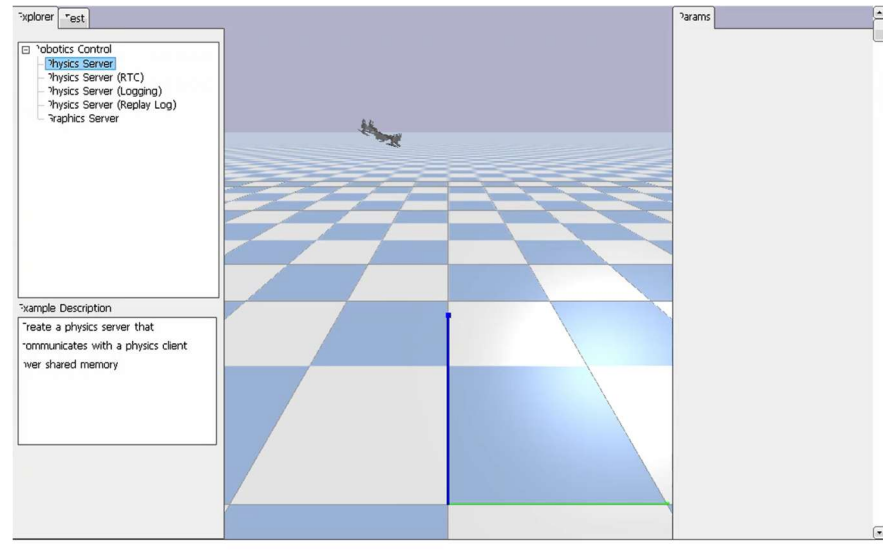


Fonte: Do Autor.

A Figura 5.15 captura o ápice da trajetória parabólica, evidenciando a fase de voo inercial do quadricóptero. Nesta etapa crítica, a aeronave encontra-se invertida em relação ao eixo horizontal, cruzando a marca de 180 graus de arfagem. Conforme validado pela otimização, o comando de empuxo dos rotores é reduzido ao seu limite mínimo. Isso evita que as hélices impulsionem o drone ativamente em direção ao chão.

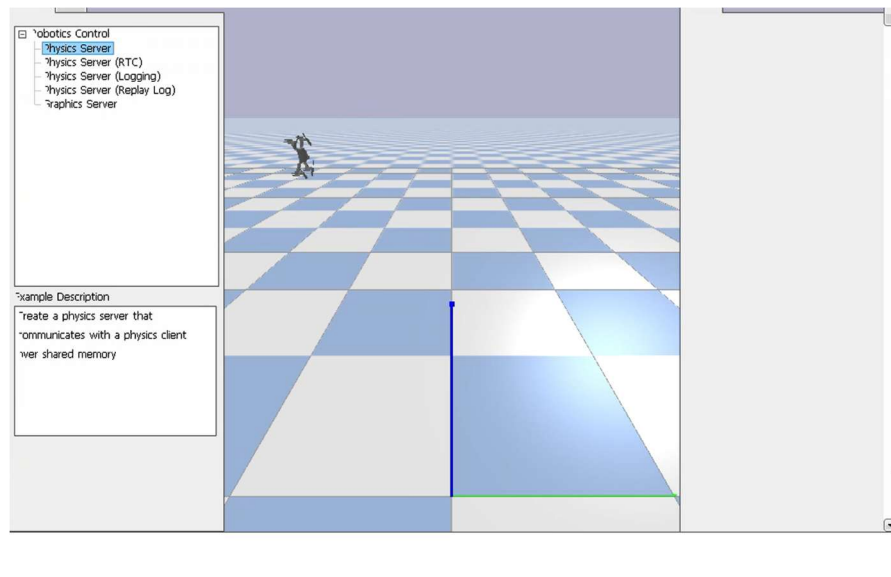
A Figura 5.16 mostra que o movimento rotacional prossegue alimentado puramente pelo momento de inércia acumulado na fase de disparo, comprovando a precisão do cálculo da dinâmica do sistema.

Figura 5.15– Ponto Intermediário (Ápice do Voo Inercial):



Fonte: Do Autor.

Figura 5.16– Ponto Intermediário em 270 graus:

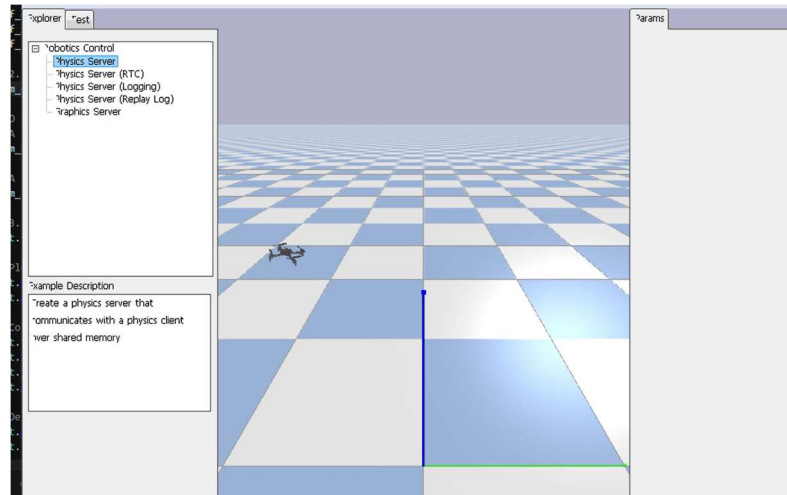


Fonte: Do Autor.

A Figura 5.17 apresenta a conclusão da manobra, no exato instante em que o veículo completa o giro de 360 graus e retoma a sua atitude nivelada. Para neutralizar a aceleração gravitacional e a velocidade de queda adquirida durante a etapa balística, o algoritmo aplica um pulso de empuxo de alta intensidade, funcionando como um freio aerodinâmico de emergência. Imediatamente após essa frenagem, a malha de controle PID convencional é restaurada,

reassumindo a autoridade sobre os motores para amortecer oscilações residuais e restabelecer a estabilidade do voo.

Figura 5.17– Ponto Final (Recuperação e Estabilização):



Fonte: Do Autor.

6 CONCLUSÕES E TRABALHOS FUTUROS

O objetivo principal deste projeto foi alcançado com sucesso: desenvolver um sistema inteligente capaz de fazer o nano-drone *Crazyfly* executar um "mortal para trás" (*backflip*) de forma autônoma e segura.

Durante o estudo, ficou claro que as fórmulas matemáticas tradicionais (como o controle PID) são excelentes para manter o drone estável em voos normais, mas falham em manobras agressivas. Isso acontece porque virar o drone de cabeça para baixo em uma fração de segundo cria turbulências no ar e forças físicas que são quase impossíveis de prever com cálculos simples.

Para resolver esse problema, a solução foi utilizar a Inteligência Artificial, especificamente a Otimização Bayesiana e os Processos Gaussianos. Em vez de tentar calcular a física exata da turbulência, o sistema usou o simulador para aprender por "tentativa e erro". O algoritmo dividiu a manobra em cinco passos simples (acelerar, iniciar o giro, deixar a inércia atuar, frear e recuperar) e descobriu os tempos exatos para ligar e desligar os motores.

Os resultados finais comprovaram que essa estratégia funciona perfeitamente. A simulação mostrou que o drone conseguiu girar 360 graus no próprio eixo e, logo em seguida, o controlador PID foi reativado para funcionar como um "freio de emergência", estabilizando a aeronave no ar e impedindo que ela capotasse ou caísse no chão.

Além de provar que é possível realizar acrobacias em robôs com poder de processamento limitado, esta pesquisa tem grande importância para o mundo real. Drones que conseguem se recuperar de posições extremas são muito mais seguros. Para o futuro desta pesquisa, sugere-se os seguintes passos:

- **Testes no Drone Real (Hardware):** Levar os tempos e parâmetros descobertos no simulador e testá-los no *Crazyfly* físico, validando como o robô lida com a carga real da bateria e o desgaste dos motores;
- **Aprender Novas Manobras:** Usar esse mesmo método de inteligência artificial para ensinar o drone a fazer outros movimentos, como giros laterais (*barrel roll*);
- **Desvio Inteligente de Obstáculos:** Conectar esse controle rápido aos sensores de distância do drone (como o módulo *MultiRanger*). Assim, o drone poderá usar movimentos rápidos e bruscos para desviar de obstáculos inesperados, o que é essencial em missões de inspeção e resgate.

REFERÊNCIAS

- ALBIERO, V.; DE BIASI, H. H. Desenvolvimento de um drone autônomo guiado por meio de técnicas de visão computacional. **Unoesc & Ciência - ACET**, Joaçaba, v. 7, n. 1, p. 35-44, jan./jun. 2016.
- ANDERSON, H.; WAXMAN, D. Autonomous Aerial Robotics: The Future of UAVs in Commercial Applications. **Journal of Autonomous Systems and Applications**, v. 10, n. 3, p. 213-229, 2019.
- ANTAL, P. **Nonlinear Control Design for Backflipping with Miniature Quadcopters**. Trabalho de Conclusão de Curso – Budapest University of Technology and Economics, Budapeste, Hungria, 2021.
- ANTAL, P.; PÉNI, T.; TÓTH, R. Backflipping With Miniature Quadcopters by Gaussian-Process-Based Control and Planning. **IEEE Transactions on Control Systems Technology**, v. 32, n. 1, p. 1-14, 2024.
- ÅSTRÖM, K. J.; HÄGGLUND, T. **Advanced PID Control**. Research Triangle Park: ISA - The Instrumentation, Systems, and Automation Society, 2006.
- BEARD, R. W.; MCLAIN, T. W. **Small Unmanned Aircraft: Theory and Practice**. Princeton: Princeton University Press, 2012.
- BITCRAZE AB. **Crazyflie 2.1**. Disponível em: <https://www.bitcraze.io/products/crazyflie-2-1/>. Acesso em: 18 ago. 2024.
- BOUABDALLAH, S.; MURRIERI, P.; SIEGWART, R. Design and control of an indoor micro quadrotor. In: **Proceedings of the IEEE International Conference on Robotics and Automation**, p. 4393-4398, 2004.
- BROCHU, E.; CORA, V. M.; DE FREITAS, N. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. **arXiv preprint** arXiv:1012.2599, 2010.
- CFD FLOW ENGINEERING. **Working principle and components of drone**. Disponível em: <https://cfdflowengineering.com/working-principle-and-components-of-drone/>. Acesso em: 10 ago. 2024.

DORF, R. C.; BISHOP, R. H. **Modern Control Systems**. 13. ed. Nova Jersey: Pearson, 2016.

EMBARCADOS. **PID digital: método de cálculo numérico**. Disponível em:

<https://embarcados.com.br/pid-digital-metodo-de-calculo-numerico/>. Acesso em: 04 set. 2024.

FAESSLER, M.; FRANCHI, A.; SCARAMUZZA, D. Differential Flatness of Quadrotor Dynamics Subject to Rotor Drag for Accurate Tracking of High-Speed Trajectories. **IEEE Robotics and Automation Letters**, v. 3, n. 2, p. 620-626, 2017.

FRAZIER, P. I. A tutorial on Bayesian optimization. **arXiv preprint** arXiv:1807.02811, 2018.

GAGEIK, N.; BENZ, P.; MONTENEGRO, S. Obstacle detection and collision avoidance for a UAV with complementary low-cost sensors. **IEEE Access**, v. 3, p. 599-609, 2015.

GIERNACKI, W. et al. Crazyflie 2.0 quadrotor as a platform for research and education in robotics and control engineering. In: **2017 22nd International Conference on Methods and Models in Automation and Robotics (MMAR)**. IEEE, p. 37-42, 2017.

KAUFMANN, E. et al. Deep drone acrobatics. In: **Robotics: Science and Systems (RSS)**, 2020.

KIM, J.; PARK, S. Infrastructure Inspection Using Autonomous Drones: Challenges and Opportunities. **International Journal of Robotics and Automation**, v. 35, n. 2, p. 145-160, 2020.

MELLINGER, D.; KUMAR, V. Minimum snap trajectory generation and control for quadrotors. In: **Proceedings of the IEEE International Conference on Robotics and Automation**, p. 2520-2525, 2011.

MICHELSON, R. C.; REECE, S. Update on flapping wing micro air vehicle research: ongoing work to develop a flapping wing, crawling entomopter. In: **13th Bristol International RPV/UAV Systems Conference Proceedings**. Bristol, Inglaterra, p. 19-24, 1998.

MOHSAN, S. A. H. et al. Unmanned aerial vehicles (UAVs): practical aspects, applications, open challenges, security issues, and future trends. **Intelligent Service Robotics**, v. 16, n. 1,

p. 109-137, 2023. Disponível em: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9841964/>. Acesso em: 11 ago. 2024.

MUEGGLER, E. et al. Aerial-guided navigation of a ground robot among movable obstacles. In: **2014 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)**. IEEE, p. 1-8, 2014.

NOGUEIRA, L. T.; BARBIERI, E. D.; RAMOS, F. M. Simultaneous Localization and Mapping (SLAM) and Data Fusion in Unmanned Aerial Vehicles: Recent Advances and Challenges. **Sensors**, v. 22, n. 7, p. 2599, 2022.

OGATA, K. **Modern Control Engineering**. 5. ed. Nova Jersey: Prentice Hall, 2010.

PALOSSSI, D. et al. An open source and open hardware deep learning-powered visual navigation engine for autonomous nano-UAVs. In: **2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)**. IEEE, p. 604-611, 2019.

SHAKHATREH, H. et al. Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges. **IEEE Access**, v. 7, p. 48572-48634, 2019.

SILVA, V. S.; MENEGHETTI, R. **Apostila de Controle I**. Departamento de Engenharia de Computação e Automação, Universidade Federal do Rio Grande do Norte. Disponível em: <https://www.dca.ufrn.br/~meneghet/FTP/Controle1/Controle%20I%20-%20Apostila.pdf>. Acesso em: 1 set. 2024.

SMITH, J.; JONES, L. Ethical and Regulatory Challenges in the Integration of Autonomous Drones into Civilian Airspace. **Journal of Aviation Law and Policy**, v. 7, n. 1, p. 54-70, 2022.



ANEXO I - Declaração sobre o Uso de Inteligência Artificial

Aluno (a): Marcio Alexandre Julieti Crochela
Matrícula: 42011ETE001
Título PFCII: Desenvolvimento de Controle Autônomo para Execução de Manobras Acrobáticas em Drones Crazyflie Utilizando Processos Gaussianos

Eu, aluno(a) com nome e matrícula supracitados, e com o respectivo trabalho de PFCII, declaro para os devidos fins que as informações prestadas a seguir refletem de forma verídica e completa o uso ou não uso de ferramentas de Inteligência Artificial neste trabalho, em conformidade com a Resolução COLCGETPM N° 19, DE 17 DE DEZEMBRO DE 2025.

SELECIONE UMA DAS OPÇÕES A SEGUIR:

☐ **NÃO HOUE USO DE FERRAMENTAS DE INTELIGÊNCIA ARTIFICIAL**

(Se marcar esta opção, não é necessário preencher os campos 1 a 4)

Declaro que não utilizei, em nenhuma etapa do desenvolvimento do presente trabalho, ferramentas de Inteligência Artificial para qualquer finalidade e assumo integral responsabilidade pelo conteúdo.

☒ **HOUE USO DE FERRAMENTAS DE INTELIGÊNCIA ARTIFICIAL**

(Preencher os campos obrigatórios abaixo, assinalando obrigatoriamente os itens 3 e 4)

Declaro que utilizei, durante o desenvolvimento do presente trabalho, as seguintes ferramentas de Inteligência Artificial e informações descritas nos itens de 1 a 4.

1. Ferramenta(s) e versão(ões) predominante(s): Gemini Pro Versão 3.1, ChatGPT Versão 5.0, Grok Versão 4.5

2. Propósito(s) (marque todos os que se aplicam):

- ☒ Exploração inicial de ideias.
- ☒ Busca de referências bibliográficas.
- ☐ Revisão de linguagem (por exemplo, melhoria de gramática ou correção de ortografia).
- ☐ Tradução técnica (conversão do texto ou partes dele para outro idioma).
- ☐ Geração automatizada de tabelas, gráficos ou figuras a partir de dados previamente analisados pelos autores.
- ☒ Programação: sugestão, depuração e documentação de códigos computacionais.

☐ Outros (especificar):



3. Autoria e validação humana:

☒ Informo que a autoria intelectual do manuscrito, bem como sua supervisão e validação, são de minha responsabilidade e que o uso de ferramentas de IA foi exclusivamente para os propósitos acima declarados.

4. Princípios éticos:

☒ Declaro que a utilização das ferramentas de IA seguiu práticas éticas, não incluindo atividades que comprometam a integridade científica tais como a geração de conteúdo original, manipulação de dados, plágio, interpretações ou análises pela IA.

☒ Declaro que respeitei direitos autorais, licenças, confidencialidade e políticas editoriais.

☒ Declaro que não enviei dados inéditos ou sensíveis à serviços que utilizam conteúdos para treinamento de modelos, exceto em plataformas institucionais ou com garantias contratuais de confidencialidade e não retenção, assegurando conformidade com a LGPD (Lei nº 13.709/2018) e demais normas de proteção de dados.

Data: 19/07/2026

Documento assinado digitalmente
gov.br MARCIO ALEXANDRE JULIETI CROCHELA
Data: 19/07/2026 18:56:52-0300
Verifique em <https://validar.iti.gov.br>

Assinatura do(a) Discente