

UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA



DISSERTAÇÃO DE MESTRADO

*Projeto de Sistema de Controle Embarcado para Controle de
Voo de Quadricópteros*

Sarah Pontes Madruga

João Pessoa - PB

Julho de 2018

Universidade Federal da Paraíba
Centro de Informática
Programa de Pós-Graduação em Informática

Projeto de Sistema de Controle Embarcado para Controle de
Voo de Quadricópteros

Sarah Pontes Madruga

Dissertação submetida à Coordenação do Curso de Pós-Graduação em
Informática da Universidade Federal da Paraíba como parte dos requisi-
tos necessários para obtenção do grau de Mestre em Informática.

Área de Concentração: Ciência da Computação
Linha de Pesquisa: Sinais, Sistemas Digitais e Gráficos

Tiago Pereira Nascimento
(Orientador)

João Pessoa, Paraíba, Brasil

©Sarah Pontes Madruga, 24 de Julho de 2018

Catálogo na publicação
Seção de Catalogação e Classificação

M183s Madrugá, Sarah Pontes.

Sistema de controle embarcado para controle de voo de quadricópteros / Sarah Pontes Madrugá. - João Pessoa, 2018.

98 f.

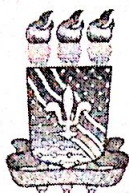
Orientação: Tiago Nascimento.

Coorientação: Alisson Brito.

Dissertação (Mestrado) - UFPB/Informática.

1. Drone, quadricóptero, VANT, controladora. I. Nascimento, Tiago. II. Brito, Alisson. III. Título.

UFPB/BC

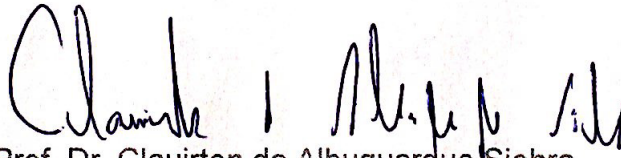


UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA



Ata da Sessão Pública de Defesa de Dissertação de Mestrado de Sarah Pontes Madruga, candidata ao título de Mestre em Informática na Área de Sistemas de Computação, realizada em 24 de julho de 2018.

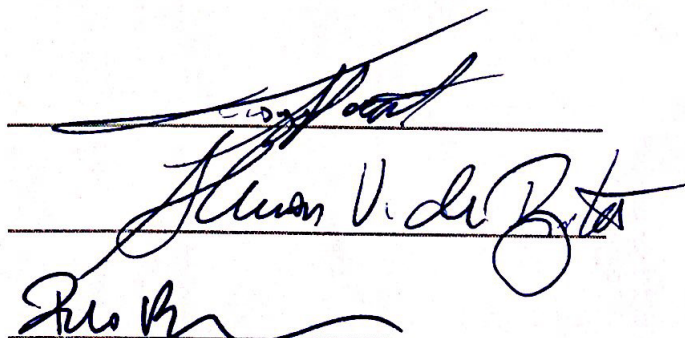
1 Aos vinte e quatro dias do mês de julho, do ano de dois mil e dezoito, às dez horas, no
2 Centro de Informática da Universidade Federal da Paraíba, em Mangabeira, reuniram-se os
3 membros da Banca Examinadora constituída para julgar o Trabalho Final da Sra. Sarah
4 Pontes Madruga, vinculada a esta Universidade sob a matrícula nº 20161020283, candidata
5 ao grau de Mestre em Informática, na área de "Sistemas de Computação", na linha de
6 pesquisa "Sinais, Sistemas Digitais e Gráficos", do Programa de Pós-Graduação em
7 Informática, da Universidade Federal da Paraíba. A comissão examinadora foi composta
8 pelos professores: Tiago Pereira Do Nascimento (PPGI-UFPB), Orientador e Presidente da
9 Banca, Alisson Vasconcelos de Brito (PPGI-UFPB), Examinador Interno,
10 Paulo Eduardo e Silva Barbosa (UEPB), Examinador Externo à Instituição. Dando início aos
11 trabalhos, o Presidente da Banca, cumprimentou os presentes, comunicou aos mesmos a
12 finalidade da reunião e passou a palavra à candidata para que a mesma fizesse a exposição
13 oral do trabalho de dissertação intitulado "Projeto de Sistema de Controle Embarcado para
14 Controle de Voo de Quadricópteros". Concluída a exposição, a candidata foi arguida pela
15 Banca Examinadora que emitiu o seguinte parecer: "**aprovada**". Do ocorrido, eu, Claurton
16 de Albuquerque Siebra, Coordenador do Programa de Pós-Graduação em Informática,
17 lavrei a presente ata que vai assinada por mim e pelos membros da banca examinadora.
18 João Pessoa, 24 de julho de 2018.


Prof. Dr. Claurton de Albuquerque Siebra

Prof. Dr. Tiago Pereira Do Nascimento
Orientador (PPGI-UFPB)

Prof. Dr. Alisson Vasconcelos de Brito
Examinador Interno (PPGI-UFPB)

Prof. Dr. Paulo Eduardo e Silva Barbosa
Examinador externo ao programa (UEPB)



Agradecimentos

A conclusão deste trabalho não seria possível sem o apoio e compreensão de diversas pessoas ao meu redor. Agradeço ao meu esposo Augusto por sempre andar ao meu lado em todas as jornadas da vida, inclusive a acadêmica. Agradeço ao meu orientador por sempre ter me apoiado na pesquisa, orientando e aconselhando da melhor forma possível. À minha família, por sempre apoiar e torcer por minhas realizações profissionais, estando presente em todos os momentos de crise e de sucesso. Por fim, a Deus, por ter me colocado em um caminho em que tudo aconteceu para dar certo neste momento, e mais ainda no futuro.

“We keep moving forward, opening new doors, and doing new things, because we’re curious, and curiosity keeps leading us down new paths.”

Walt Disney

Resumo

Neste trabalho, foi desenvolvido um projeto de sistema embarcado para uma controladora de voo aplicada a quadricópteros. Com componentes acessíveis, arquitetura e software abertos, ele torna possível a implementação e teste de diversos algoritmos de controle pelo usuário, diferenciando-se das alternativas hoje existentes no mercado. Para tal, há um sensoriamento para a captura e registro da odometria do drone, os comandos são passados para os motores via Modulação por Largura de Pulso, e o processamento é feito em uma Raspberry Pi 3.

Para validação da arquitetura, foram feitos diversos experimentos em cada componente do sistema, separados e em conjunto, analisando sua aplicabilidade ao objetivo desejado, além de um estudo da modelagem matemática e da estrutura de um quadricóptero. Ainda, testou-se a sintonia de um controlador Proporcional Integral Derivativo para verificar o funcionamento do dispositivo desenvolvido.

Palavras-chave: Sistemas embarcados, drone, quadricóptero, VANT, controladora.

Abstract

In this project, we developed an embedded system for a flight cotroller applied to quadrotor UAVs. Built with accessible components, open hardware architecture and open software, it makes possible the tests and implementation of new control algorithms by the user, setting it apart from the most common alternatives available on the market. Therefore, there is a sensing system for capturing and recording the quadrotor odometry, the angular velocity commands are sent to the motors via PWM, and the processing is made on a Raspberry Pi 3.

For architecture validation, experiments were performed on each component, individually and together with the whole system, analysing then the aplicability to the desired goal. Furthermore, a study of the mathematical modeling and structure of a quadcopter was conducted. Lastly, the tuning of a Proportional Integral Derivative controller was tested to verify the operation of the developed device.

Keywords: Embedded systems, drone, quadrotor, UAV, flight controller.

Sumário

Lista de Figuras	vii
Lista de Tabelas	ix
Lista de Códigos Fonte	x
1 Introdução	1
1.1 Motivação	1
1.2 Objetivos	3
1.3 Contribuições	3
1.4 Metodologia	4
1.5 Estrutura da Dissertação	4
2 Fundamentação Teórica	6
2.1 Modelagem Matemática de um Quadricóptero	6
2.1.1 Modelo Cinemático	8
2.1.2 Modelo Dinâmico	11
2.2 Sistemas Embarcados	13
2.3 Estratégia de Controle PID	15
2.4 Algoritmo de Sintonia Baseado em PSO	18
2.5 Considerações Finais	20
3 Trabalhos Relacionados	21
3.1 Trabalhos relacionados sobre Sistemas Embarcados aplicados a Controladores	21
3.2 Trabalhos relacionados sobre Sistemas Embarcados aplicados a Drones . .	22
3.3 Considerações Finais	24

4	Projeto do Sistema de Controle	25
4.1	Componentes Básicos do Sistema de um Quadricóptero	25
4.2	Componentes da Controladora e Sensoriamento	29
4.2.1	Sensoriamento	29
4.2.2	Módulo PWM	29
4.2.3	Conexão dos Componentes	30
4.3	Sistema de Controle	32
4.4	Considerações Finais	35
5	Avaliação Experimental	36
5.1	Simulações do Método de Sintonia Baseado em PSO	36
5.2	Sensoriamento	39
5.2.1	Giroscópio e Acelerômetro	41
5.2.2	Barômetro	41
5.3	Módulo de Comandos PWM	44
5.4	Testes de Sintonia do Controlador na Raspberry Pi	47
5.5	Considerações Finais	49
6	Conclusão	50
	Referências Bibliográficas	57
A	Patente <i>Hardware para Controladora Customizável para Quadricópteros</i>	58
B	Patente <i>Gaiola de Testes para Quadricópteros</i>	59

Lista de Figuras

2.1	Esquema de eixos em "+", além de detalhamento sobre variáveis pertinentes para o modelo do quadricóptero.	7
2.2	Configuração de eixos em "x" para o quadricóptero.	8
2.3	Diagrama de blocos descrevendo o funcionamento de um controlador PID. .	17
2.4	Exemplo de uma malha de controle completa para um quadricóptero.	18
4.1	Drone utilizado para os experimentos	26
4.2	Detalhe da controladora conectada ao quadricóptero.	26
4.3	Motores utilizados no sistema de propulsão. Os de rosca prata giram no sentido anti-horário, e os de rosca preta, no sentido horário.	27
4.4	ESCs utilizados para passar comandos aos motores.	27
4.5	Módulo de sensoriamento GY-86.	29
4.6	Módulo PWM Adafruit PCA9685.	30
4.7	Esquema de conexão de todos os componentes do quadricóptero.	31
4.8	Raspberry Pi 3.	31
4.9	Sensor ultrassônico HC-SR04.	32
4.10	Detalhe frontal e traseiro da placa de circuito impresso.	32
4.11	Projeto da PCB.	33
4.12	Componentes da controladora.	33
4.13	Conexão da PCB com a Raspberry Pi e todos os outros módulos.	33
4.14	Controladora completa.	34
4.15	Esquema do funcionamento do controle com os sensores.	34
5.1	Comparação das trajetórias realizadas com a sintonia tradicional (esquerda) e com o PSO (direita).	37

5.2	Comparação dos gráficos em 3D para a sintonia tradicional (esquerda) e com o PSO (direita).	37
5.3	Comparação das trajetórias realizadas com a sintonia tradicional (esquerda) e com o PSO (direita).	38
5.4	Comparação dos gráficos em 3D para a sintonia tradicional (esquerda) e com o PSO (direita).	39
5.5	Dados de <i>roll</i> , <i>pitch</i> e <i>yaw</i> coletados durante o teste.	42
5.6	Velocidades angulares P , Q , R	42
5.7	Representação gráfica da altitude derivada dos dados coletados pelo barômetro.	44
5.8	Representação gráfica dos dados coletados na Tabela 5.3.	45
5.9	Saídas do controlador PID referentes ao motor M2 enviadas ao módulo PWM.	48
5.10	Comandos de velocidade angular enviados ao motor M2 pelo módulo PWM.	48

Lista de Tabelas

2.1	Influência das velocidades dos rotores no movimento do quadricóptero . . .	8
5.1	Análise de Erro Médio Quadrático	39
5.2	Ganhos calculados e usados nas simulações	40
5.3	Dados coletados no teste de resposta aos comandos PWM	46

Lista de Códigos Fonte

5.1	Código para coletar dados do barômetro e calcular a altitude	43
-----	--	----

Capítulo 1

Introdução

Neste capítulo, é feita uma breve introdução do trabalho, contextualizando-o e analisando seus objetivos. São expostas, também, a motivação para o estudo do tema e as contribuições desta pesquisa. Ao final, examina-se a estrutura desta dissertação, citando brevemente o que é encontrado em cada capítulo da leitura.

1.1 Motivação

No decorrer da última década, Veículos Aéreos Não Tripulados (VANT) ou *Unmanned Aerial Vehicles* (UAV) têm sido vastamente utilizados para facilitar a vida humana. Eles podem desempenhar missões difíceis ou de risco, por exemplo, de cunho militar [Fahlstrom e Gleason 2012], ou tornar tarefas civis mais eficientes, como entrega de mercadorias.

Até 2023, espera-se que o mercado de drones para aplicações comerciais atinja 26,74 bilhões de dólares, a uma taxa de crescimento anual de 20,7% de 2016 a 2023 [MRC 2017]. A utilização de drones também deve aumentar mais ainda nesse período, por uso civil e militar [Mazur, Wisniewski e McMillan].

Nesse contexto, a configuração em forma de quadricóptero é de bastante interesse, pois permite decolagem e pouso verticais, além de pairar em áreas pequenas, o que é impossível para veículos de asa fixa. A utilização de quatro rotores também é mais eficiente que o tradicional helicóptero com apenas uma hélice, pois as quatro hélices menores do quadricóptero reduzem o torque do sistema como um todo [Hussein e Nemah 2015].

A despeito de sua configuração mecânica simples, seu comportamento dinâmico torna o

controle da aeronave substancialmente difícil, pois trata-se, de fato, de um sistema subatuado não linear [Bouktir, Haddad e Chettibi 2008]. Ou seja, há apenas quatro rotores para gerar impulsos de entrada, a fim de controlar os seis graus de liberdade do drone. Em outras palavras, quatro comandos de entrada para seis variáveis de saída.

Ainda, quadricópteros são considerados classicamente como sistemas dinâmicos instáveis [Hussein e Nemah 2015]. Então, o estudo de algoritmos de controle adequados para atingir a estabilidade é imprescindível, especialmente em uma área em relação à qual o interesse da comunidade científica só cresceu nos últimos anos.

Apesar da relevância e do aumento do estudo de algoritmos de controle para VANTs, ainda não se encontra facilmente no mercado uma plataforma para experimentos reais acessível e adaptável, em que o usuário possa aplicar seus próprios algoritmos. A vasta maioria das alternativas disponíveis, apesar de muitas serem de software aberto, não permitem a implementação de um novo algoritmo de controle, apenas a sintonização de um já existente, por causa da maneira como foram desenvolvidas [Ebeid, Skriver e Jin 2017], [Ebeid et al. 2018]. Esses dispositivos foram criados para aqueles usuários que querem usar o quadricóptero, não necessariamente pesquisar e testar sobre ele, por isso possuem as características mencionadas.

Dada a natureza da quantidade de tarefas que o drone tem de executar, a fim de operar normalmente e desempenhar mesmo suas funções mais básicas, é necessário ter um conjunto de dispositivos atuando de maneira integrada e independente, e que os dados coletados (posicionamento, orientação, ou quaisquer outros sensores essenciais para a especificidade da tarefa a ser realizada) por eles sejam processados. Consequentemente, tem-se a necessidade de um sistema embarcado no VANT para lidar com tudo isso, garantindo sua autonomia.

Esse trabalho propõe, então, uma arquitetura aberta para uma controladora de voo para quadricópteros, em que nenhum dos componentes funciona como uma caixa preta, com o objetivo de customizar os experimentos o máximo possível de acordo com as necessidades do pesquisador ou usuário.

1.2 Objetivos

O objetivo central desta pesquisa é o projeto de um sistema embarcado para uma controladora de voo aplicada a quadricópteros. Este dispositivo deve ter componentes acessíveis e de baixo custo, arquitetura e software abertos, facilitando a pesquisa e teste de algoritmos de controle diferentes, diferenciando-se das alternativas mais comuns existentes no mercado, as quais funcionam como "caixa preta", quando se trata do controlador.

Mais especificamente, busca-se montar um sensoramento para a captura e registro da odometria do drone; identificar uma maneira confiável de passar comandos via Modulação por Largura de Pulso (*PWM - Pulse Width Modulation*) para os motores; e mostrar um design prático e eficiente para unir todos os componentes por meio de uma placa de circuito impresso (*PCB - Printed Circuit Board*).

Ainda, o conjunto dos resultados obtidos deve ser submetido a publicação em conferências e periódicos relevantes para a área, além das inovações tecnológicas, que, por sua vez, devem ser protegidas mediante o depósito de patentes.

1.3 Contribuições

Como principais contribuições desta pesquisa, podemos destacar o principal objetivo: o desenvolvimento de uma controladora de voo com algoritmo de controle 100% modificável pelo usuário e com componentes acessíveis e de baixo custo.

Ao longo do estudo, foi desenvolvido também um método de sintonia para controladores baseado em Otimização por Enxame de Partículas (*PSO - Particle Swarm Optimization*), que levou à publicação de um artigo em conferência.

Ainda, projetou-se uma estrutura de testes para controladores de quadricópteros, com seis graus de liberdade, denominada Gaiola de Testes para Quadricópteros, que foi submetida em forma de patente.

Esses e outros resultados relevantes deste trabalho foram compilados em forma de artigos e depósito de patentes, ratificando a relevância da pesquisa:

- MADRUGA S.; BASSO G.; TAVARES A.; BRITO A.; NASCIMENTO T.; A PSO-based Tuning Algorithm for Quadcopter Controllers. In: *2018 XXII Brazilian Confe-*

rence in Automation. (Aceito, a ser apresentado em setembro de 2018).

- MADRUGA S.; TAVARES A.; BRITO A.; NASCIMENTO T.; A Project of an Embedded Control System for Quadrotor UAVs. In: *2018 VIII Brazilian Symposium on Computing Systems Engineering*. (Submetido).
- Patente depositada no INPI: HARDWARE PARA CONTROLADORA CUSTOMIZÁVEL PARA QUADRICÓPTEROS. Patente de Invenção (PI). Número do processo: BR 10 2018 009992 2 (Apêndice A)
- Patente depositada no INPI: GAIOLA DE TESTES PARA QUADRICÓPTEROS. Patente de Invenção (PI). Número do processo: BR 10 2018 010097 1 (Apêndice B)

1.4 Metodologia

Primeiramente, foi feito um estudo do modelo dinâmico de um quadricóptero, e a caracterização dos parâmetros reais do drone foi feita no laboratório. O controlador para esse modelo foi simulado no Simulink[®], para assim possibilitar melhor entendimento das características e do comportamento do objeto de estudo. Mais detalhes serão abordados no Capítulo 2.

Logo após, realizou-se uma ampla pesquisa sobre os tipos de arquitetura disponíveis para a implementação de controladores em drones, a fim de decidir o esboço de um protótipo. Do mesmo modo, para sua montagem, foi necessário investigar os sensores e componentes, disponíveis no mercado, mais acessíveis e com desempenho satisfatórios.

Cada sensor precisou ser estudado e testado separadamente, antes do teste conjunto com todos os componentes. Em seguida, foram feitos experimentos para sintonizar o controlador, utilizando a arquitetura finalizada e, assim, validando-a. Os procedimentos realizados para testes e validação são discutidos em mais detalhes no Capítulo 5.

1.5 Estrutura da Dissertação

No Capítulo 2 deste trabalho, abordam-se os conhecimentos teóricos fundamentais para compreender como funciona um quadricóptero, como sua física, dinâmica e modelagem

computacional. Em seguida, um pouco de teoria de controle Proporcional Integral Derivativo (PID), escolhida para exemplificar a aplicação do dispositivo apresentado. Por fim, propõe-se um método de sintonia para os ganhos do controlador, baseado em PSO.

No Capítulo 3, discutem-se os trabalhos relacionados encontrados na literatura, comparando o que já foi estudado e publicado na área com os resultados obtidos.

No Capítulo 4, estuda-se a fundo a estrutura do drone e, em seguida, o detalhamento de cada parte da arquitetura e do sensoriamento.

No Capítulo 5, constam o relatório e a análise dos experimentos realizados ao longo da pesquisa, incluindo a exposição dos resultados e concluindo na validação da arquitetura.

No Capítulo 6, são apresentadas as conclusões gerais do trabalho.

Nos Apêndices, ao final do trabalho, é possível encontrar uma cópia dos documentos das patentes resultantes da pesquisa aqui desenvolvida. Códigos relevantes utilizados para os testes de controle e sensoriamento que não apareceram ao longo do texto podem ser encontrados em um repositório *GitHub*, referenciado como [Madruga 2018].

Capítulo 2

Fundamentação Teórica

Neste capítulo, é apresentado um resumo dos modelos cinemático e dinâmico de um quadricóptero, dado que a modelagem é uma parte muito extensa do estudo e não era o foco deste trabalho. No entanto, a compreensão de certos aspectos físicos é indispensável para que se possa construir um sistema para o controle de voo que leve em consideração as especificidades relativas ao seu comportamento, e também para uma boa interpretação dos resultados. Este tópico será detalhado na Seção 2.1 .

Na Seção seguinte, a 2.2, são introduzidos os conceitos fundamentais de Sistemas Embarcados relevantes para o trabalho aqui desenvolvido. Na Seção 2.3, discute-se a lógica de controle Proporcional Integral Derivativo (PID), a qual foi usada para validar a arquitetura para controle de voo proposta neste trabalho. Finalmente, na Seção 2.4, aborda-se a técnica de sintonia baseada em Enxame de Partículas (PSO) desenvolvida para encontrar ganhos ótimos para o controlador descrito na Seção 2.3.

2.1 Modelagem Matemática de um Quadricóptero

Primeiramente, vamos abordar alguns conceitos básicos mostrando a ideia principal do modelo dinâmico de um quadricóptero, descrevendo intuitivamente os movimentos necessários para seu voo.

O drone é formado por uma estrutura rígida, como pode ser visto nas Figuras 2.1 e 2.2. Em cada ponta há um motor com hélice, e o eixo de rotação delas é fixo e paralelo. Nessa disposição, considera-se que a única variação relevante é a velocidade de rotação

das hélices [Bresciani 2008]. Observa-se também, na Figura 2.2, que os motores opostos giram no mesmo sentido, que é contrário à rotação do outro par. Isso equilibra o torque do sistema como um todo e elimina a necessidade de um rotor traseiro, como na configuração tradicional do helicóptero [Bresciani 2008]. As Figuras 2.1 e 2.2 também detalham as duas configurações possíveis de eixos cartesianos para a caracterização do drone.

Ratifica-se que, apesar de o quadricóptero ter seis graus de liberdade, temos apenas quatro rotores para os quais enviar comandos, ou seja, trata-se de um sistema subatuado. As variáveis selecionadas para melhor controlar o sistema são: *Throttle* ("quantidade de aceleração"), *Roll* ("balanço" ou "rolagem"), *Pitch* ("arfagem") e *Yaw* ("guinada"). Os termos entre parênteses estão em tradução livre.

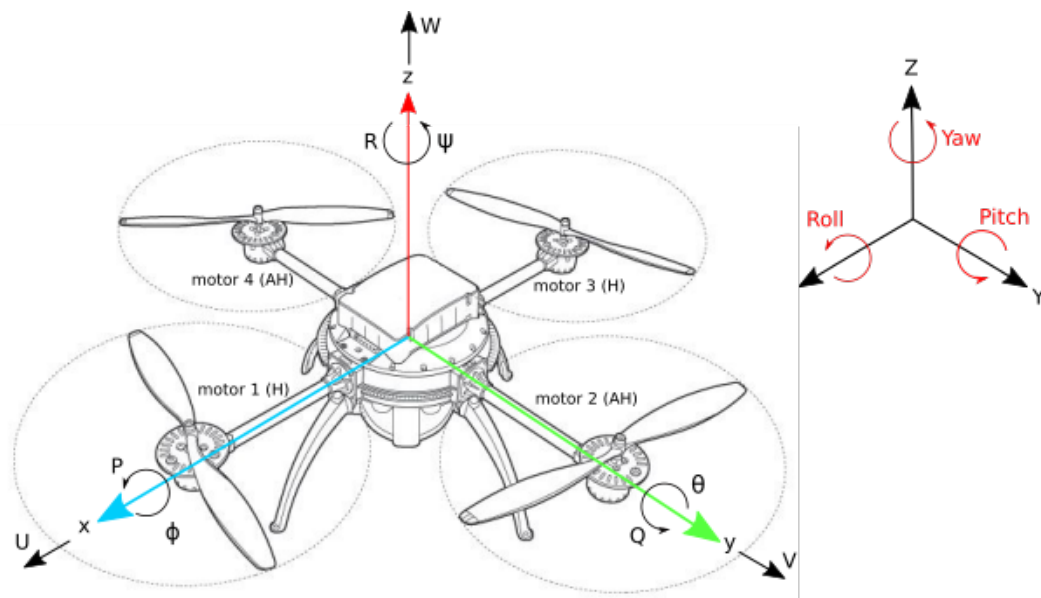


Figura 2.1: Esquema de eixos em "+", além de detalhamento sobre variáveis pertinentes para o modelo do quadricóptero.

O comando *Throttle* nada mais é que o aumento ou diminuição das velocidades dos quatro motores na mesma quantidade. Ele é responsável pela alteração na altitude do VANT.

O movimento de *Roll* é caracterizado pela rotação em torno do eixo x , que pode ser atingida aumentando a velocidade dos rotores (ou do rotor, no caso de uma configuração em "+") à direita do eixo, e diminuído a dos rotores à esquerda, ou vice-versa, fazendo o drone mover-se lateralmente.

O *Pitch* é muito similar ao *Roll*, a diferença é que ele faz o drone mover-se para frente

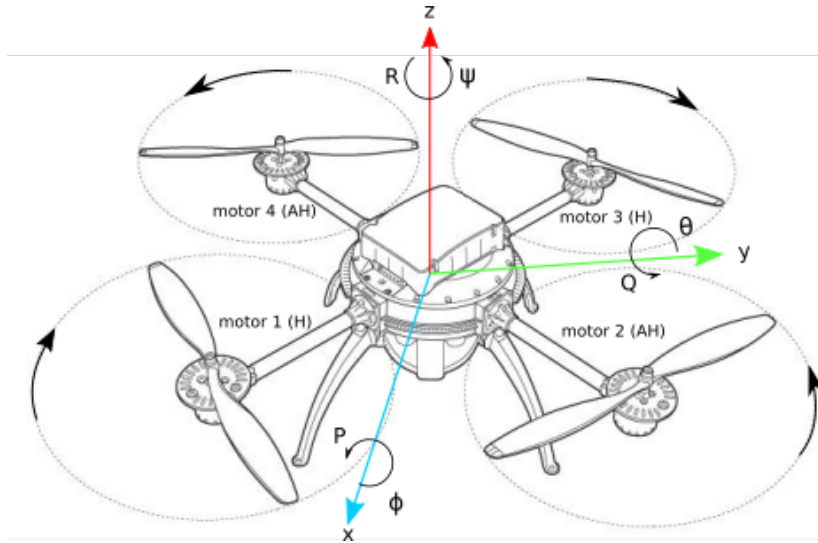


Figura 2.2: Configuração de eixos em "x" para o quadricóptero.

ou para trás, segundo o referencial. Ou seja, trata-se de uma rotação em torno do eixo y . Para tal, analogamente, aumenta-se a velocidade dos rotores (ou do rotor, no caso de uma configuração em "+") à direita do eixo, e diminui-se a dos rotores à esquerda.

O *Yaw* define a rotação em torno do eixo z . Para este comando, deve-se aumentar (ou diminuir) a velocidade do par de rotores girando no sentido horário, e diminuir (ou aumentar) a velocidade do par em sentido anti-horário.

A Tabela 2.1 resume os conceitos explicados acima [Hussein e Nemah 2015].

	ω_1	ω_2	ω_3	ω_4
z (<i>Throttle</i>)	↑	↑	↑	↑
ϕ (<i>Roll</i>)	↑	-	↓	-
θ (<i>Pitch</i>)	-	↑	-	↓
ψ (<i>Yaw</i>)	↓	↑	↓	↑

Tabela 2.1: Influência das velocidades dos rotores no movimento do quadricóptero

2.1.1 Modelo Cinemático

Para a representação matemática do modelo cinemático do quadricóptero, dois conjuntos de eixos são necessários para referência: o sistema de coordenadas do robô e o sistema de coordenadas da Terra, que podem ser vistos na Figura 2.1. O primeiro é fixo, enquanto o

segundo move-se, linear e angularmente, de acordo com o drone, e são necessárias matrizes de transformação para converter as variáveis de um sistema para outro.

Considerando a regra da mão direita, podemos destacar três matrizes de rotação individuais em torno de cada eixo: $R(x, \phi)$, na Equação 2.1, representa a rotação em torno do eixo x ; $R(y, \theta)$, na Equação 2.2, representa a rotação em torno do eixo y ; $R(z, \psi)$, na Equação 2.3, representa a rotação em torno do eixo z [Bouabdallah 2007].

$$R(x, \phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \quad (2.1)$$

$$R(y, \theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \quad (2.2)$$

$$R(z, \psi) = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

A matriz de rotação completa $R(\phi, \theta, \psi)$ é o produto das três rotações anteriores, e significa três rotações sucessivas em torno de cada eixo respectivo. Assim, temos a Equação 2.4 [González-Sánchez et al. 2013]:

$$R(\phi, \theta, \psi) = R(x, \phi)R(y, \theta)R(z, \psi) \quad (2.4)$$

Que resulta em:

$$R = \begin{bmatrix} \cos \psi \cos \theta & \cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi & \cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi \\ \sin \psi \cos \theta & \sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi & \sin \psi \sin \theta \cos \phi - \sin \phi \sin \psi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \quad (2.5)$$

A matriz exposta na Equação 2.5 é usada na conversão das coordenadas do sistema de eixos do robô $(\vec{x}, \vec{y}, \vec{z})$ para o sistema de eixos da Terra $(\vec{X}, \vec{Y}, \vec{Z})$, ou seja, se um ponto em

$(\vec{x}, \vec{y}, \vec{z})$ sofre três rotações sucessivas, ele pode ser representado em $(\vec{X}, \vec{Y}, \vec{Z})$ pela Equação 2.6:

$$r_{X,Y,Z} = R(\phi, \theta, \psi) \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad (2.6)$$

De onde podemos tirar as Equações 2.7, 2.8 e 2.9 que, derivadas em relação ao tempo, resultam nas correspondentes velocidades.

$$\begin{aligned} r_X(x, y, z) = & (\cos \psi \cos \theta)x + (\cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi)y \\ & + (\cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi)z \end{aligned} \quad (2.7)$$

$$\begin{aligned} r_Y(x, y, z) = & (\sin \psi \cos \theta)x + (\sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi)y \\ & + (\sin \psi \sin \theta \cos \phi - \sin \phi \sin \psi)z \end{aligned} \quad (2.8)$$

$$r_Z(x, y, z) = (-\sin \theta)x + (\cos \theta \sin \phi)y + (\cos \theta \cos \phi)z \quad (2.9)$$

De maneira análoga, as derivadas dos ângulos de Tait-Bryan (ϕ, θ, ψ) , no eixo da Terra, relacionam-se com as velocidades angulares (P, Q, R), no eixo do robô, segundo a Equação 2.10 [Bouabdallah 2007], [Mellinger 2012]:

$$\begin{bmatrix} P \\ Q \\ R \end{bmatrix} = R_r \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (2.10)$$

Onde:

$$R_r = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \sin \phi \cos \theta \\ 0 & -\sin \phi & \cos \phi \cos \theta \end{bmatrix} \quad (2.11)$$

Quando os ângulos (ϕ, θ, ψ) são pequenos, a Equação 2.10 pode ser aproximada pela

identidade [Paiva et al. 2016]:

$$\begin{bmatrix} P \\ Q \\ R \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (2.12)$$

Assim, a cinemática de um corpo rígido com 6 graus de liberdade pode ser representada por:

$$\dot{\xi} = J_{\theta}\nu \quad (2.13)$$

Onde: $\dot{\xi} = [\dot{X} \ \dot{Y} \ \dot{Z} \ \dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T$ e $\nu = [U \ V \ W \ P \ Q \ R]^T$, e J_{θ} é a matriz de conversão detalhada na Equação 2.14:

$$J_{\theta} = \begin{bmatrix} R & 0_{3 \times 3} \\ 0_{3 \times 3} & R_r \end{bmatrix} \quad (2.14)$$

2.1.2 Modelo Dinâmico

Para o modelo dinâmico, começaremos definindo a matriz de momentos de inércia J_b , na Equação 2.15. Essa matriz representa os momentos de inércia do quadricóptero em cada eixo, e é de extrema relevância para a dinâmica de voo do sistema [Mahony, Kumar e Corke 2012].

$$J_b = \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix} \quad (2.15)$$

J_b é a inércia do drone relativa ao sistema de eixos do robô, e J_{xx} , J_{yy} and J_{zz} são a inércia através de cada eixo x , y e z desse sistema, respectivamente.

O empuxo T de um único conjunto motor/hélice pode ser calculado pela Equação 2.16, onde C_T é o coeficiente de empuxo para um rotor específico, ρ é a densidade do ar, A_r é a área da superfície de rotação da hélice, r é o raio do rotor e ω é a velocidade angular do rotor.

$$T = C_T \rho A_r r^2 \times \omega^2 \quad (2.16)$$

Para simplificar a identificação dos parâmetros, as constantes podem ser agrupadas em um só termo, como mostra a Equação 2.17, onde c_T representa as constantes citadas acima, e ω , a velocidade angular do rotor.

$$T = c_T \times \omega^2 \quad (2.17)$$

Analogamente, é possível provar a Equação 2.18 para o torque Q do sistema motor/hélice. Nesse caso, c_Q é o coeficiente de torque, e ω ainda é a velocidade angular do rotor.

$$Q = c_Q \times \omega^2 \quad (2.18)$$

Com tais informações, pode-se demonstrar a Equação 2.19, que mostra a matriz que descreve os empuxos e torques no sistema completo do quadricóptero [Mahony, Kumar e Corke 2012], para uma configuração em "+", onde d é a distância entre os motores e seus respectivos eixos de rotação, ou seja, o comprimento do braço do drone do centro até o motor, neste caso.

$$\begin{bmatrix} \Sigma T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} c_T & c_T & c_T & c_T \\ 0 & d \cdot c_T & 0 & -d \cdot c_T \\ -d \cdot c_T & 0 & d \cdot c_T & 0 \\ -c_Q & c_Q & -c_Q & c_Q \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \quad (2.19)$$

Para uma configuração de eixos em "x", temos a Equação 2.20, na qual $d_x = d \sin(45^\circ)$, pois essa seria a distância entre o motor/hélice e o eixo de rotação. Nessa configuração, o efeito de c_Q não muda, enquanto c_T vai afetar todos os quatro motores, tanto no *pitch* (θ) quanto no *roll* (ϕ).

$$\begin{bmatrix} \Sigma T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} c_T & c_T & c_T & c_T \\ -d_x \cdot c_T & d_x \cdot c_T & d_x \cdot c_T & -d_x \cdot c_T \\ -d_x \cdot c_T & -d_x \cdot c_T & d_x \cdot c_T & d_x \cdot c_T \\ -c_Q & c_Q & -c_Q & c_Q \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \quad (2.20)$$

A seguir, na Equação 2.21, destacamos a aproximação linear da relação de Comando de Throttle [Hartman et al.]. Aqui, ω_{ss} é a velocidade esperada, em RPM, do motor no regime permanente; ($Throttle\%$) é o comando de *throttle* em porcentagem; c_R é o coeficiente de

conversão de *throttle* para RPM; e b é o ponto de cruzamento no eixo y , na regressão linear [Hartman et al.].

$$\omega_{ss} = (Throttle\%)c_R + b \quad (2.21)$$

Para a matriz final, também é necessário levar em consideração as forças giroscópicas resultantes no corpo, que são governadas pela inércia de cada componente rotatório do motor (J_m); pelas velocidades angulares de *roll* e *pitch* (P e Q); e pela velocidade de cada conjunto motor/hélice (ω_i). Isso posto, as Equações 2.22 e 2.23 mostram o torque giroscópico criado pelos motores nas ações de *roll* e *pitch*.

$$\tau_{\phi_{gyro}} = J_m Q \left(\frac{\pi}{30} \right) (\omega_1 - \omega_2 + \omega_3 - \omega_4) \quad (2.22)$$

$$\tau_{\theta_{gyro}} = J_m P \left(\frac{\pi}{30} \right) (-\omega_1 + \omega_2 - \omega_3 + \omega_4) \quad (2.23)$$

Finalmente, para a configuração em "+", a matriz final pode ser vista na Equação 2.24, que se refere aos momentos presentes no quadricóptero, resultantes da aerodinâmica, empuxos e torques no sistema.

$$M = \begin{bmatrix} d.c_T.\omega_2^2 - d.c_T.\omega_4^2 + J_m Q \left(\frac{\pi}{30} \right) (\omega_1 - \omega_2 + \omega_3 - \omega_4) \\ -d.c_T.\omega_1^2 + d.c_T.\omega_3^2 + J_m P \left(\frac{\pi}{30} \right) (-\omega_1 + \omega_2 - \omega_3 + \omega_4) \\ -c_Q.\omega_1^2 + c_Q.\omega_2^2 - c_Q.\omega_3^2 + c_Q.\omega_4^2 \end{bmatrix} \quad (2.24)$$

2.2 Sistemas Embarcados

Todos os anos, bilhões de sistemas de computadores são construídos para um propósito bem diferente do que estamos acostumados. Eles não são os nossos conhecidos computadores pessoais, ou PCs. Esses sistemas são embarcados em outros dispositivos eletrônicos, repetidamente cumprindo uma função específica, e muitas vezes passando despercebidos pelo usuário.

Apesar de ser difícil estabelecer uma definição específica, esse tipo de sistema pode ser mais bem compreendido por meio de exemplos e análise de suas características [Vahid e

Givargis 2001]. Sistemas embarcados podem ser encontrados em uma variedade de dispositivos comuns do nosso dia a dia: celulares, fornos micro-ondas, termostatos, copiadoras, impressoras, injeção eletrônica de automóveis, VANTs.

Sistemas embarcados possuem uma série de características em comum:

- *Função Única*: Um sistema embarcado normalmente executa apenas um programa, repetidamente. Ou, em algumas exceções, programas diferentes, mas com objetivos afins. Um micro-ondas é um micro-ondas, e uma controladora de voo é uma controladora de voo, mesmo com funções extras adicionadas, o funcionamento sempre vai ser associado ao programa principal. Diferentemente de um PC, que pode executar programas editores de texto, jogos etc.
- *Alta Restrição*: Restrições são comuns em todo tipo de sistema computacional. Os embarcados, no entanto, enfrentam desafios maiores. O *design* deve ser elaborado levando em conta uma série de características. O ideal é que o custo seja o mais baixo possível, o tamanho, muitas vezes, precisa ser reduzido para caber em um *chip*, o desempenho deve ser rápido o suficiente para se obter dados em tempo real, e o consumo de potência deve ser mínimo, para economizar bateria e evitar aquecimento excessivo.
- *Reativos e em Tempo Real*: muitos sistemas embarcados devem reagir constantemente a mudanças no ambiente e no comportamento do sistema e, assim, fornecer respostas em tempo real. Por exemplo, a controladora de voo deve monitorar a odometria do quadricóptero e fornecer, sem atraso, os dados ao programa que executa o algoritmo de controle, como mudança na postura, na altitude etc. Uma falha nessa resposta poderia levar a erros no sistema.

Além de cumprir sua funcionalidade, a implementação de um sistema embarcado deve otimizar uma série de características de *design* mensuráveis. Entre as inúmeras existentes, podemos citar: o custo para fabricar cada unidade, o custo de projetar o sistema, tamanho, desempenho, potência, possibilidade de flexibilizar as funções do sistema sem aumentar o custo de projeto, segurança, confiabilidade nos resultados.

É importante notar que cada uma dessas características normalmente são otimizadas em detrimento de outra. Por exemplo, diminuir o tamanho do dispositivo pode prejudicar o

desempenho ou aumentar o custo de fabricação. As métricas mais importantes a serem aprimoradas dependem de cada projeto e de suas restrições.

O sistema de controle embarcado projetado durante esta pesquisa tem as seguintes características:

- Fácil acessibilidade de componentes;
- O tamanho está condizente com as alternativas do mercado;
- Desempenho em tempo real;
- O consumo de bateria decorre principalmente do funcionamento dos motores. O sistema da controladora necessita apenas de uma alimentação de 5V;
- O sistema é flexível e aceita alterações no algoritmo pelo usuário;
- O sistema é tão seguro quanto qualquer quadricóptero, após a sintonização do controlador;
- A confiabilidade dos resultados depende unicamente da qualidade do algoritmo de controle empregado. Foram adicionados filtros aos sensores embutidos no sistema, para filtragem de ruídos.

2.3 Estratégia de Controle PID

A técnica de controle Proporcional Integral Derivativo (PID) é largamente utilizada em quadricópteros, sobretudo para o controle de pose ou postura (*roll*, *pitch* e *yaw*). O controle PID clássico ou com variações pode ser encontrado na grande maioria das controladoras de voo comerciais [Ebeid, Skriver e Jin 2017], [Ebeid et al. 2018], e também na literatura recente relacionada ao tema [Paiva et al. 2016], [Paiva et al. 2015], [García F.R. Rubio 2012], [Lukmana e Nurhadi 2015], [Abbas e Sami 2017].

Assim, esse controlador foi escolhido como um exemplo de aplicação possível da arquitetura e do sensoriamento propostos neste trabalho. Contudo, enfatiza-se que o dispositivo aqui projetado permite que o algoritmo de controle seja desenvolvido livremente pelo usuário.

O controlador PID lê os dados dos sensores e calcula a velocidade de rotação ideal dos motores para que o quadricóptero responda corretamente aos comandos. O objetivo é corrigir o erro, ou seja, a diferença de um valor medido (normalmente por um giroscópio) e um valor desejado. Esse erro pode ser minimizado ajustando as entradas de controle (velocidade de rotação dos motores) a cada iteração, para que assim as saídas (*roll*, *pitch*, *yaw* e *throttle*) sejam as desejadas.

O controlador proporcional ajusta a velocidade dos motores adicionando o valor do erro, $e(t)$, que precisa ser multiplicado por um ganho Kp para ser convertido ao valor adequado. Veja a Equação 2.25. O ganho Kp determina a responsividade do controlador para corrigir o erro. Generalizando, Kp alto indicaria um controle mais brusco, enquanto Kp baixo provocaria uma resposta mais suave.

Quando o valor do erro é muito alto ou quando a resposta precisa ser ajustada com rapidez, o controlador proporcional provoca *overshoots*, começa a compensar o erro além do necessário, e o drone começa a se comportar de maneira errática. Por isso, adiciona-se o termo derivativo.

$$Ajuste = Kp \times e(t) \quad (2.25)$$

O controlador derivativo observa a variação do erro, criando um erro maior se a variação está muito rápida ou um menor se estiver muito lenta para o sistema. Isso ajuda a suavizar a taxa de correção e a prevenir mudanças erráticas de velocidade. Se o ganho Kd estiver muito elevado, o drone pode sofrer com vibrações, pois este controlador amplifica o ruído no sistema. Além disso, um Kd excessivo também provoca um atraso na resposta do quadricóptero aos comandos.

Um controle Proporcional Derivativo (PD) tem um bom desempenho, mas pode não entregar uma consistência de velocidade ao longo do tempo. A Equação 2.26 mostra o termo derivativo do controlador.

$$Ajuste = Kd \times \frac{de(t)}{dt} \quad (2.26)$$

O controle integral ajuda no desempenho em regime permanente, ajustando o controlador para pequenas mudanças nos erros. Ele faz isso por meio de uma soma de todos os erros anteriores, multiplicada por uma constante Ki , observe a Equação 2.27.

O ganho K_i determina o quanto o controlador de voo vai trabalhar para manter a pose do drone contra forças externas, como vento e centro de gravidade deslocado. Se o ganho K_i ficar muito alto, o quadricóptero parecerá insensível aos comandos.

$$Ajuste = K_i \times \int e(t) \times dt \quad (2.27)$$

A expressão completa para um controlador PID pode ser vista na Equação 2.28, e a Figura 2.3 detalha o seu funcionamento por meio de um diagrama de blocos.

$$Ajuste = K_p \times e(t) + K_i \times \int e(t) \times dt + K_d \times \frac{de(t)}{dt} \quad (2.28)$$

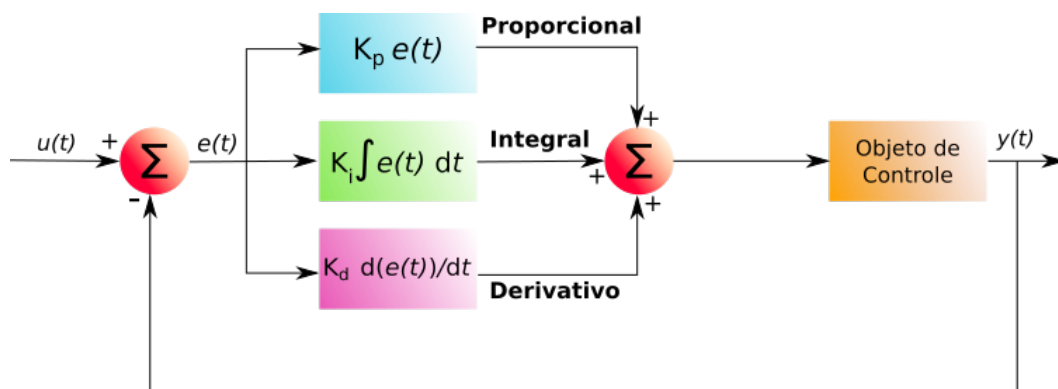


Figura 2.3: Diagrama de blocos descrevendo o funcionamento de um controlador PID.

A Figura 2.4 mostra o diagrama de blocos para um sistema de controle completo, de pose e trajetória, para um quadricóptero autônomo. O esquema é composto de um *loop* externo, que controla a trajetória que o drone deve seguir; e de um *loop* interno, que controla a pose que deve ser obedecida para o seguimento do caminho desejado.

Primeiramente, o bloco *Gerador de Trajetória* envia os comandos de caminho ou posição (x, y) para o *Controlador de Trajetória*. Este controlador tem como saída os comandos de *roll*, *pitch* e *yaw* para o *Controle de Pose*. O comando de altitude (z ou *throttle*) vem diretamente do *Gerador de Trajetória*.

É importante notar que, no *Controlador de Trajetória*, há na realidade dois controladores, um para os comandos em x , e outro para os comandos em t . O mesmo ocorre dentro do bloco *Controle de Pose e Altitude*, há um controlador para a altitude z , um para o *roll* ϕ , um para o *pitch* θ e um para o *yaw* ψ .

As respectivas correções nas velocidades dos motores para provocar o comportamento desejado em ϕ , θ , ψ e z serão então enviadas ao quadricóptero, que deve possuir o sensoriamento adequado para uma boa *Realimentação de Estados* e, conseqüentemente, um bom cálculo dos erros nos próximos *loops* de controle.

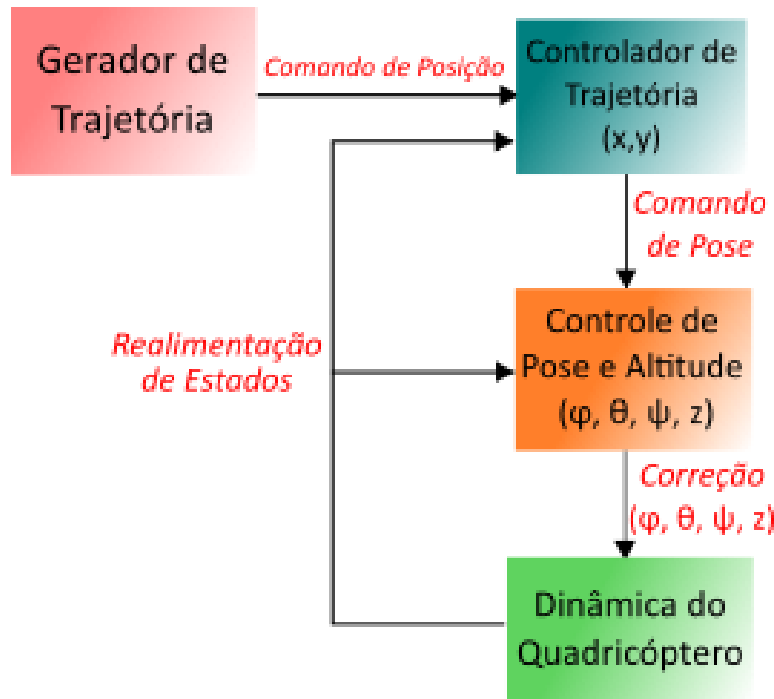


Figura 2.4: Exemplo de uma malha de controle completa para um quadricóptero.

Os ganhos Kp , Ki e Kd precisam ser bem ajustados para cada planta ou objeto de controle, tanto em simulação quanto experimentalmente. Na próxima seção, propõe-se um método de sintonia em testes computacionais, e, no Apêndice B, detalha-se uma gaiola de testes desenvolvida e patenteada com o objetivo de sintonizar controladores para drones em testes experimentais.

2.4 Algoritmo de Sintonia Baseado em PSO

Nosso método de sintonia consiste de três etapas. Primeiramente, executa-se uma busca em enxame de partículas (PSO) pelos melhores valores para os ganhos do controlador, durante simulações do voo do drone, usando uma adaptação do modelo contido em [Hartman et al.]. Cada combinação de parâmetros testada é salva, junto com o erro de trajetória resultante, com o objetivo de mapear as combinações de ganhos com o erro.

Os dados salvos são usados para modelar o Erro de Trajetória (TE) como uma função dos parâmetros, ou seja, modela-se o erro como uma função de alta dimensão dos ganhos do controlador. Com esse modelo, faz-se uma busca, no espaço de parâmetros, pelo menor erro, usando novamente PSO. Assim, evita-se o custo computacional excessivo da simulação de voo, executando a busca num construto matemático da própria simulação.

Para maior clareza, o passo a passo é descrito a seguir:

- *Otimização por Enxame de Partículas (PSO):*

Foi usado o método de PSO proposto em [Wang e Liu 2016] e testado em [Basso et al. 2017]. Diferentemente de algoritmos de PSO padrão, em que as partículas são guiadas apenas pelo melhor conjunto de parâmetros históricos e pelos melhores parâmetros do enxame, é introduzida a informação do melhor vizinho, um mecanismo de abandono e um operador de busca caótica.

- *Modelar TE como uma função dos ganhos do controlador:*

Durante a execução do PSO, buscam-se várias combinações de ganhos diferentes, cada uma com uma efetividade diferente no rastreamento de trajetória. Cada conjunto de parâmetros é salvo, junto com seu Erro de Trajetória (TE). Essa informação é usada para modelar o erro como uma função multidimensional dos ganhos do controlador.

Assume-se que TE é uma função dos ganhos do tipo polinomial esparsa de alta ordem, assim, toma-se uma expansão de Taylor truncada e acham-se os pesos da expansão adequados. Esse passo é modelado como mostrado na Equação 2.29:

$$TE = P \times w \quad (2.29)$$

Onde TE representa o Erro de Trajetória, P é o vetor contendo cada combinação polinomial de ganhos, até a ordem desejada; e w é o vetor com os pesos da expansão de Taylor.

O problema remanescente é resolvido empregando um algoritmo de minimização ℓ^0 , como demonstrado em [Mohimani, Babaie-Zadeh e Jutten 2009].

- *Segunda Otimização por Enxame de Partículas (PSO):*

Com o modelo calculado, realiza-se então mais um PSO para a obtenção dos ganhos ótimos para o controlador.

Mais detalhes sobre a execução deste método de sintonia, bem como os resultados dos experimentos com a simulação de duas trajetórias, serão abordados no Capítulo 5.

2.5 Considerações Finais

Este capítulo apresenta conceitos fundamentais para a compreensão do restante do trabalho e dos experimentos realizados. São mostrados os significados de cada uma das quatro variáveis usadas para controlar e modelar o VANT, o *roll*, o *pitch*, o *yaw* e o *throttle*.

Expõe-se a modelagem cinemática e dinâmica do drone, ainda que de maneira resumida, pois não se tratava do foco principal da pesquisa, sendo apenas necessária uma compreensão básica. Ainda assim, fica clara a sua complexidade física. Destaca-se também as possíveis configurações de eixos que podem ser usadas para modelar o quadricóptero.

São apresentados os conceitos mais pertinentes, para este contexto, sobre Sistemas Embarcados. Ao final, o sistema de controle de voo desenvolvido nesta pesquisa é caracterizado segundo os critérios expostos.

São explicadas, analogamente, as bases da teoria de controle PID, que foi escolhida para ser implementada como forma de exemplificar a aplicação da controladora. Da mesma forma, detalham-se as funções e efeitos de cada termo e de cada ganho multiplicativo no desempenho do voo, e um método é proposto de sintonia computacional, a fim de estabelecer valores ótimos para tais ganhos.

Capítulo 3

Trabalhos Relacionados

Neste capítulo, discutem-se os trabalhos relacionados aos temas relevantes para esta pesquisa, analisando o estado da arte, estudando o que já foi publicado na literatura em forma de artigos e comparando com tecnologias já disponíveis no mercado.

A discussão foi dividida em duas partes. A primeira, na Seção 3.1, analisa trabalhos publicados em que há aplicação de controladores a projetos em geral, utilizando sistemas embarcados para experimentos práticos e verificação do funcionamento. Em seguida, abordam-se projetos de sistemas embarcados aplicados a drones, seja para controladores, seja para processamento de dados, bem como suas aplicações.

3.1 Trabalhos relacionados sobre Sistemas Embarcados aplicados a Controladores

A aplicação de um controlador a qualquer objeto de controle envolve naturalmente o uso de sistemas embarcados. Por isso, há inúmeras pesquisas envolvendo experimentos com controladores que mencionam alguma arquitetura embarcada utilizada.

Podemos citar diversos exemplos recentes, como [Obayes, Al-Saedi e Mohammed 2017], que utilizam uma Raspberry Pi e um Arduino para desenvolver um protótipo de controlador sem fio para uma máquina fresadora; [Anduray e Irigoyen 2017] também utilizam uma Raspberry Pi, a fim de implementar um controlador para controlar o nível de líquidos; em [Ate e Abdelrahim 2018], a Raspberry Pi é novamente usada, desta vez para um sistema de controle

de temperatura. Como se pode observar, a Raspberry Pi é uma alternativa bastante popular para a implementação de controladores, pois tem boa capacidade de processamento e possui sistema operacional Linux.

Não obstante, é possível encontrar na literatura exemplos de controladores que foram desenvolvidos em outras plataformas, muitas vezes não especificadas na pesquisa. Um controlador para um sistema de irrigação alimentado a energia solar, utilizando um microcontrolador KL25Z, é apresentado em [Eragamreddy e Sree 2017]. [Iyer et al. 2017] mostram um controlador embarcado para controlar a vazão de gases.

Aplicações à área de robótica podem ser vistas em [Wang e Wang 2016] e [Li, Wang e Yi 2015], que propõem arquiteturas de controle embarcado para robôs quadrúpedes; [Zheng et al. 2015] mostram a implementação de um controle embarcado para um sistema de locomoção com 6 graus de liberdade; e [Pal, Shekhar e Sharma 2009] projetam e implementam um controlador de velocidade para aplicações na área de micromanufaturas, como impressoras 3D ou fresadoras.

3.2 Trabalhos relacionados sobre Sistemas Embarcados aplicados a Drones

Um quadricóptero ou qualquer tipo de VANT tem por característica básica a necessidade de um sistema que atue de maneira independente e processe dados das mais diversas naturezas, sejam eles informações necessárias para o seu funcionamento, como dados de sensoriamento e controle, sejam resultados de uma missão, como imagens coletadas ou dados de temperatura. Afinal, um drone não vai voar atrelado fisicamente a uma unidade não embarcada (um computador, por exemplo, um PC).

Em vista disto, pesquisas relacionadas a Sistemas Embarcados com aplicação para esse tipo de veículo são imprescindíveis. Essencialmente, sempre que há um experimento prático com VANTs, há algum tipo de sistema embarcado envolvido. Apresenta-se, nesta seção, o estado da arte da pesquisa mais recente e mais relevante ao nosso contexto.

Já há bastantes plataformas de controladoras de voo para quadricópteros disponíveis no mercado, como foi mostrado em [Ebeid, Skriver e Jin 2017] e [Ebeid et al. 2018]. Entre elas, podemos citar principalmente a linha *PixHawk* e a *APM*, baseadas em Arduino, muito

conhecidas entre os usuários e algumas das mais utilizadas. O grande problema dessas placas é o funcionamento fechado do controlador. Apesar de um ajuste nos ganhos ser possível, o algoritmo de controle não pode ser alterado, trata-se de um PID em que apenas Kp , Ki e Kd podem ser modificados. Isso torna a pesquisa de novos algoritmos de controle da área de VANTs complicada, pois os dispositivos mais acessíveis e com mais documentação disponível para a aplicação não podem ser utilizados para esse objetivo.

Algumas soluções começaram a surgir muito recentemente (dentro do último ano), porém elas não são necessariamente muito acessíveis. O MatLab[®] disponibilizou, na última versão de janeiro de 2018, um suporte integrado para dois quadricópteros da linha PARROT, o Mambo Minidrone e o AR.Drone [MathWorks], [MathWorks], prometendo o *design* de algoritmos de controle com o auxílio do Simulink[®]. Apesar de bastante atraente, essa alternativa está restrita a um tipo específico de drone, de uma determinada marca, e pode-se provar bastante custosa financeiramente. Seu lançamento recente, no entanto, mostra a relevância e a necessidade de plataformas do tipo para testes de algoritmos de controle em VANTs.

As abordagens de sistemas embarcados direcionadas para drones não são somente para controladoras de voo. Há diversos trabalhos com soluções embarcadas para processamento de imagem e vídeo com as mais variadas aplicações, como pode ser visto em [Mhatre et al. 2015], com processamento de vídeo, ou em [Silva et al. 2015], que propõem um sistema de processamento de imagens aéreas para identificar falhas em plantações. [Tijtgat et al. 2017] e [Smyczynski, Starzec e Granosik 2017] apresentam técnicas de detecção de objetos utilizando arquiteturas embarcadas em VANTs, o primeiro com um algoritmo de detecção executando em uma NVIDIA Jetson TX2, e o segundo, com uma Pixhawk como controladora de voo e uma Raspberry Pi 3 como computador embarcado.

Outras aplicações podem ser encontradas em [Yarlequé, Alvarez e Martínez 2017], em que um radar foi montado em um VANT para estudos arqueológicos, com uma Raspberry Pi 2 para processar os dados. [Schuetz et al. 2017] também propõem o uso de drones para sensoriamento remoto com radares. [Benitez et al. 2016] desenvolveram um protótipo de VANT para a inspeção visual de linhas elétricas aéreas, utilizando uma APM como controladora de voo.

O sensoriamento necessário para monitoramento da odometria do quadricóptero também

é outro objeto de estudo pertinente. [Astudillo et al. 2017] mostram um controlador PID funcionando em um Arduino Mega, com o sensoriamento feito utilizando um *smartphone* Nexus 4. Uma solução com fusão de sensores e Filtro de Kalman testada em um sistema embarcado foi proposta em [Hetényi, Gótzky e Blázovics 2016].

Por fim, existem algumas propostas de controladoras de voo desenvolvidas para diversas plataformas. Em [Monterrosa et al. 2016], vemos uma proposta de controladora desenvolvida usando um sistema embarcado baseado em FPGA; em [Luo et al. 2014], propõe-se um sistema de controle de navegação de um quadricóptero para ambientes internos, usando uma APM como controladora de voo e uma Raspberry Pi para processamento de dados; [Solidakis et al. 2017] sugerem um sistema baseado em Arduino para controlar VANTs usando GSM. Em [Ebeid, Skriver e Jin 2017], é mencionada uma controladora baseada em Raspberry Pi em que o algoritmo de controle pode ser customizado, chamada *Erle-Brain*. Todavia, além de dificilmente haver acesso a essa plataforma aqui no Brasil, mesmo no exterior ela tem um preço bastante elevado, custando em torno de duzentos euros.

3.3 Considerações Finais

Neste capítulo, foram citados alguns trabalhos relacionados ao objeto de estudo, utilizados para formar uma base de conhecimento para esta pesquisa. Proucurou-se estabelecer a relação entre controladores e sistemas embarcados de maneira clara, além de mostrar a relevância de sua aplicação a quadricópteros.

Quando comparado às alternativas citadas, o projeto proposto nesta dissertação possui como diferencial a acessibilidade, pois é feito com componentes de fácil aquisição. Outro ponto de interesse é o fato de o sistema ser todo implementado numa Raspberry Pi, plataforma baseada em Linux com boa capacidade de processamento e customização de algoritmo, além da conexão simples a diversos tipos de sensores, câmeras etc.

Evidentemente, não é possível mencionar toda a literatura disponível sobre o tema. É feita, portanto, uma seleção do que há de mais recente e expressivo para o desenvolvimento do trabalho. Assim, o leitor deve considerar a bibliografia aqui disponibilizada como um ponto de partida para o entendimento do assunto.

Capítulo 4

Projeto do Sistema de Controle

Neste capítulo, encontra-se uma descrição da estrutura básica de um quadricóptero, mencionando os equipamentos utilizados no drone desta pesquisa. A seguir, faz-se uma listagem, acompanhada de uma explicação, dos componentes de *hardware* que integram o projeto da controladora cuja patente pode ser vista no Apêndice A. Na última seção, é mostrado um detalhamento de como o sistema de controle foi implantado na plataforma escolhida, destacando o papel de cada uma de suas partes.

4.1 Componentes Básicos do Sistema de um Quadricóptero

Aqui explicaremos os principais componentes de um sistema autônomo para um quadricóptero, ou seja, um sistema de *hardware* e *software* capaz de substituir o guia humano durante o voo [Ebeid, Skriver e Jin 2017]. Na Figura 4.1, observa-se o drone construído para os experimentos deste trabalho.

- *Controladora de Voo*

É a principal placa de processamento de informações e de envio de comandos do VANT. Segundo o algoritmo desenvolvido pelo usuário, ela controla os motores e se comunica com os sensores que transmitem as informações necessárias. Neste projeto, foi utilizada uma Raspberry Pi 3 para executar o algoritmo de controle e interpretar os dados do sensoriamento conectado a ela. Detalhe da controladora projetada com todas as devidas conexões feitas pode ser visto na Figura 4.2.



Figura 4.1: Drone utilizado para os experimentos

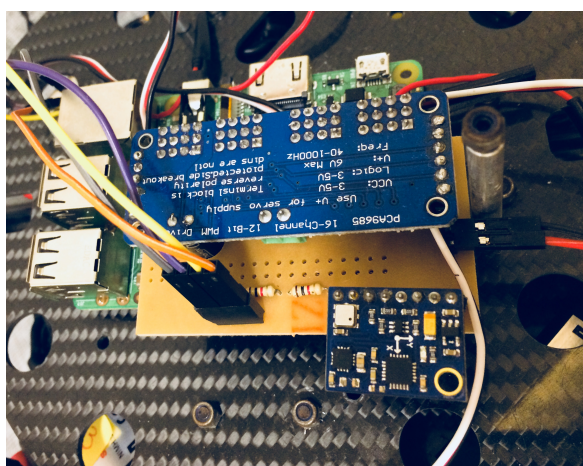


Figura 4.2: Detalhe da controladora conectada ao quadricóptero.

- *Sistema de Propulsão*

Para VANTs com vários rotores, normalmente são utilizados motores DC do tipo *brushless*, controlados por ESCs (*Electronic Speed Controller*), que enviam os comandos de velocidade de rotação para os motores. Neste trabalho, foram utilizados motores EMAX MT3510 600kV, retratados na Figura 4.3; e ESCs Simonk 30A, Figura 4.4.



Figura 4.3: Motores utilizados no sistema de propulsão. Os de rosca prata giram no sentido anti-horário, e os de rosca preta, no sentido horário.



Figura 4.4: ESCs utilizados para passar comandos aos motores.

- *Sensores*

Os sensores são necessários para detectar mudanças durante o voo ou mesmo para coletar dados durante a missão. Inúmeros tipos de sensores podem ser conectados ao sistema, dependendo da funcionalidade desejada. Elegemos os sensores mais utilizados na comunidade para implementar neste projeto.

O Módulo de Medição Inercial (IMU - *Inertial Measurement Unit*) é o principal componente de navegação do VANT. Possui um acelerômetro e um giroscópio para estimar a pose do quadricóptero, ou seja, os valores de *roll*, *pitch* e *yaw* em tempo real necessários para a realimentação do controlador e cálculo do erro, como explicado anteriormente, na Seção 2.3.

O Barômetro é usado para medir a pressão atmosférica e calcular a altitude do drone, podendo também fornecer a temperatura ambiente. Os mais sensíveis conseguem detectar mudanças centimétricas.

O Magnetômetro é um instrumento usado para medir a intensidade, direção e sentido de campos magnéticos em sua proximidade. Pode ser utilizado como bússola, sendo muitas vezes aplicado na fusão dos dados de vários sensores, a fim de promover uma maior precisão dos resultados.

Mais detalhes sobre o *hardware* de sensoriamento podem ser encontrados na próxima seção.

- *Sistema de Comunicação*

O sistema de comunicação com o VANT pode ser de dois tipos: uma Estação de Controle Terrestre ou um Transmissor a Rádio. O primeiro, escolhido neste projeto, é uma unidade computacional (neste caso, um PC) que se comunica com o quadricóptero via protocolo SSH, sem fios, e monitora seu funcionamento. A segunda opção é utilizada quando um piloto humano controla o drone via transmissor a rádio (ou controle remoto), ou seja, quando o VANT não é autônomo. Importante observar que, mesmo neste segundo caso, não se pode voar sem uma plataforma controladora de voo.

4.2 Componentes da Controladora e Sensoriamento

4.2.1 Sensoriamento

Para o sensoriamento do quadricóptero, foi utilizado o módulo GY-86, constituído por:

- IMU MPU6050, que contém um acelerômetro de 3 eixos e um giroscópio de 3 eixos.
- Barômetro MS5611, sensor de pressão com precisão capaz de detectar diferenças de centímetros na altitude.
- Magnetômetro HMC5883L com 3 eixos.

O módulo GY-86 pode ser visto na Figura 4.5.

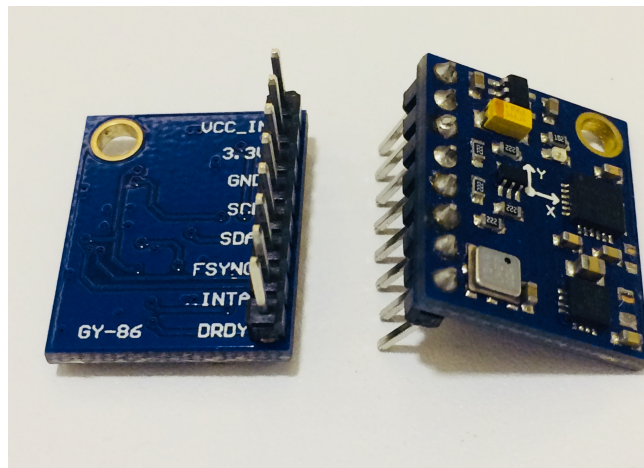


Figura 4.5: Módulo de sensoriamento GY-86.

4.2.2 Módulo PWM

Para que os comandos de velocidade cheguem aos motores, eles precisam passar pelos ESCs, ou seja, o controlador deve enviar seus comandos para os ESCs. Estes, por sua vez, recebem os comandos via modulação por largura de pulso (PWM - *Pulse Width Modulation*).

A Raspberry Pi tem a capacidade de enviar sinais PWM. No entanto, o seu PWM é gerado por *software*, o que o torna ligeiramente impreciso para certas aplicações. Por isso, escolhemos usar, para enviar pulsos PWM aos ESCs, o módulo Adafruit PCA9685 que, além de gerar os sinais com precisão, possui 16 canais, ou seja, com ele seria possível controlar

até VANTs com mais de quatro rotores, pois é necessário apenas um canal para cada motor. Este dispositivo pode ser visto em destaque na Figura 4.6.

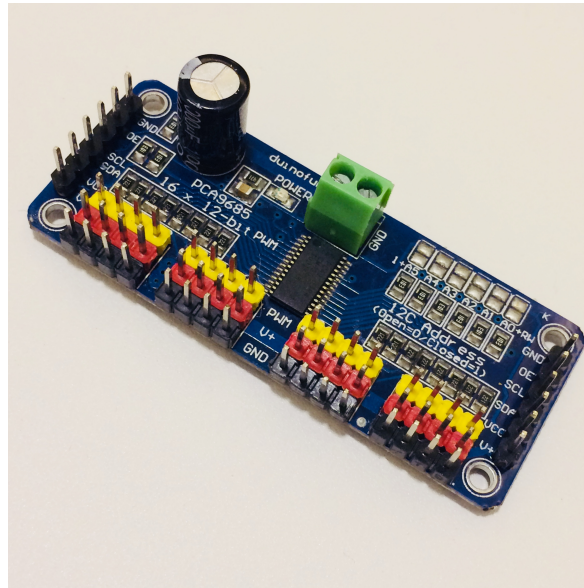


Figura 4.6: Módulo PWM Adafruit PCA9685.

4.2.3 Conexão dos Componentes

Com os componentes apresentados nas seções anteriores, a seguir mostraremos como são feitas as conexões para a montagem da controladora, como mostra a Figura 4.7.

Como já foi mencionado, utilizamos uma Raspberry Pi 3 para o processamento dos dados, mostrada na Figura 4.8. O módulo de sensores GY-86 e o módulo PWM PCA9685 comunicam-se com esta placa por meio do protocolo I²C, assim, eles devem conectar-se aos pinos específicos para esse tipo de conexão na Raspberry Pi 3, especificados na Figura 4.7. Os ESCs são conectados cada um a um canal da PCA9685, e, em seguida, aos motores. A alimentação do sistema é fornecida por uma bateria.

O esquema de conexões da Figura 4.7 ainda propõe a adição de um sensor ultrassônico HC-SR04, que não está sendo utilizado aqui, mas decidimos deixar a possibilidade de conexão a cargo do usuário, pois é um sensor muito popular, que possui inúmeras aplicações, por exemplo, detecção de objetos ou obstáculos. Detalhe deste sensor na Figura 4.9.

Para uma conexão simples e segura de todos os componentes, projetamos uma placa de circuito impresso (PCB - *Printed Circuit Board*), segundo as conexões mostradas na Figura

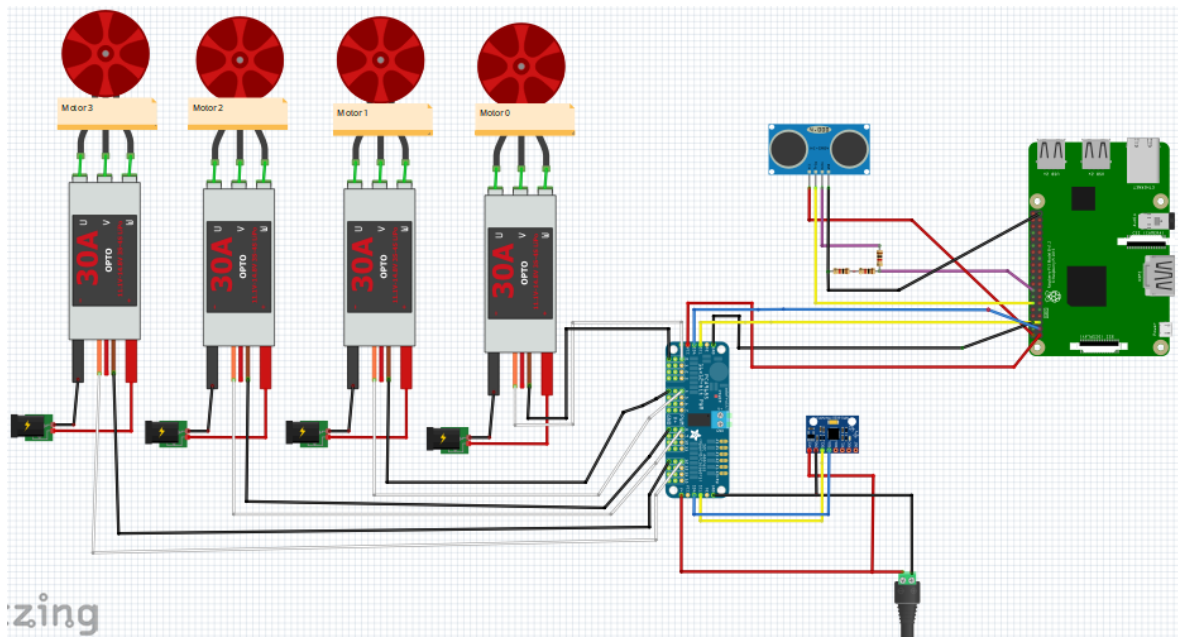


Figura 4.7: Esquema de conexão de todos os componentes do quadricóptero.



Figura 4.8: Raspberry Pi 3.

4.7, a qual pode ser vista na Figura 4.10, e seu projeto, na Figura 4.11.

A Raspberry Pi conecta-se à parte traseira da PCB, enquanto os outros sensores vão na parte frontal, como de costume. Tem-se assim um esquema de conexão em "sanduíche" que, apesar de não ortodoxo quando se trata de PCBs, provou-se bastante seguro e estável quando testado junto às vibrações naturais do quadricóptero. Observe a Figura 4.13.

Todas as partes que constituem a controladora podem ser vistas na Figura 4.12.

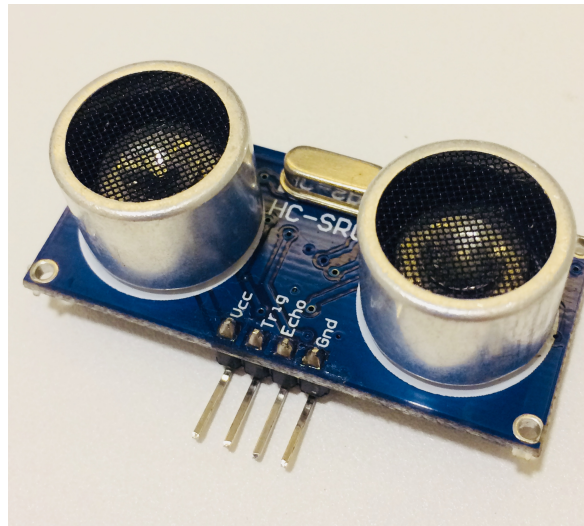


Figura 4.9: Sensor ultrassônico HC-SR04.

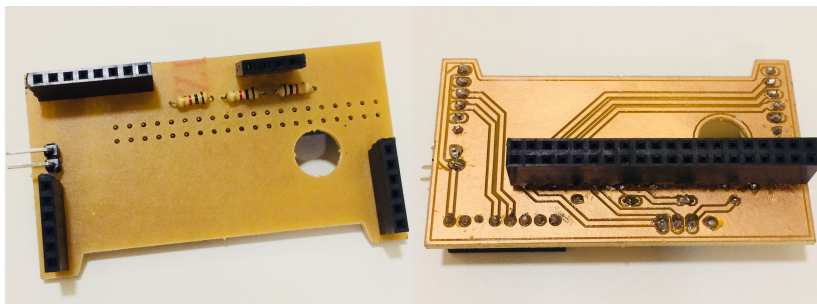


Figura 4.10: Detalhe frontal e traseiro da placa de circuito impresso.

4.3 Sistema de Controle

Como exemplo de aplicação da controladora desenvolvida, mostrada completa na Figura 4.14, o controle de pose e altitude, com o detalhamento dos sensores, pode ser visto na Figura 4.15.

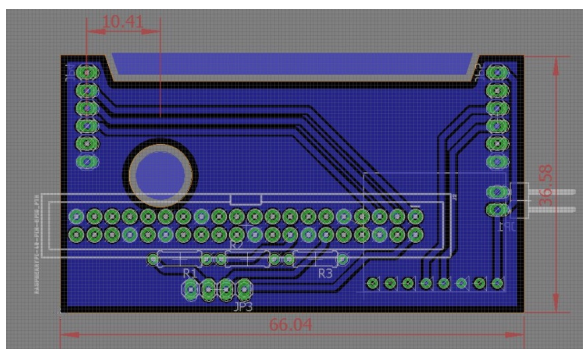


Figura 4.11: Projeto da PCB.



Figura 4.12: Componentes da controladora.

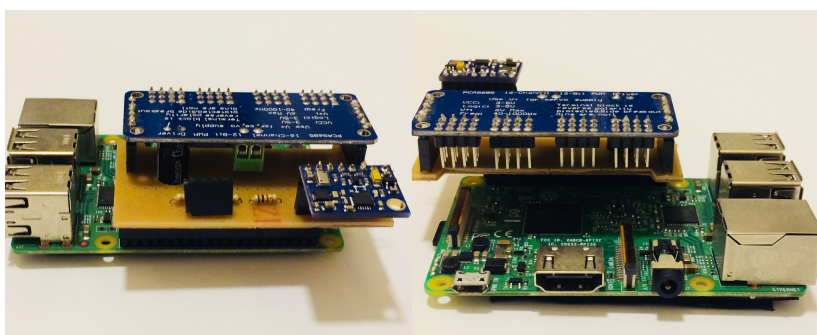


Figura 4.13: Conexão da PCB com a Raspberry Pi e todos os outros módulos.

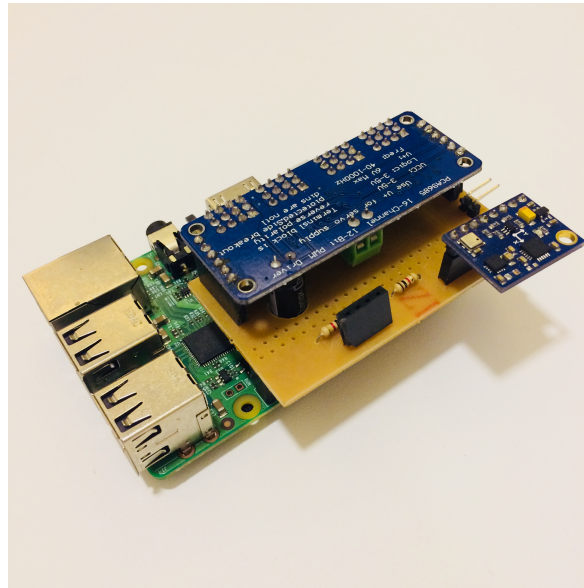


Figura 4.14: Controladora completa.

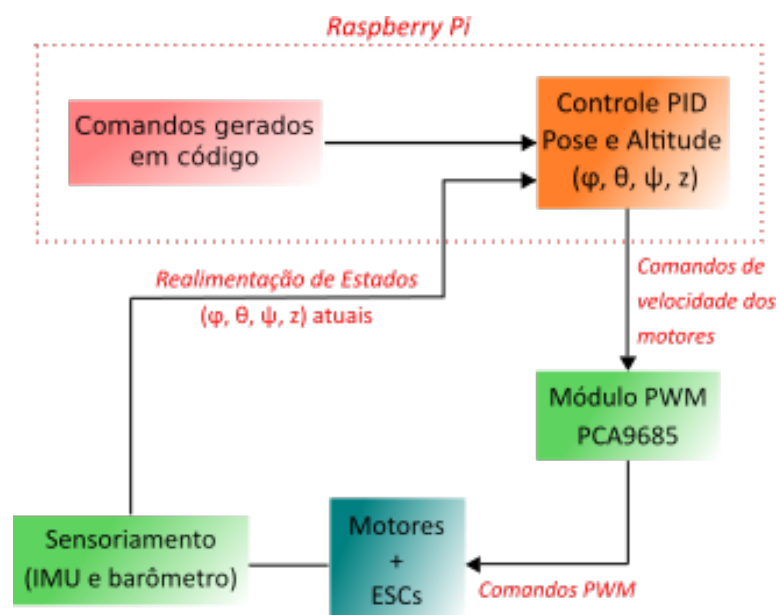


Figura 4.15: Esquema do funcionamento do controle com os sensores.

Visando a empregar o dispositivo em um drone autônomo, isto é, que voa sem um piloto humano, os comandos de *roll*, *pitch*, *yaw* e altitude são gerados em código, na Raspberry Pi, e enviados para o controlador. Este, por sua vez, usa os dados recebidos do sensoramento para calcular o valor do erro entre o comando e o estado atual do drone. As velocidades adequadas dos motores são assim transmitidas para o módulo PWM, que as envia para o conjunto motor/ESC. Os sensores encaminham uma atualização dos estados do quadricóptero para o controlador, e o *loop* recomeça e continua até o funcionamento ser interrompido.

Detalhes de código podem ser encontrados em [Madruga 2018].

4.4 Considerações Finais

Neste capítulo, discutiram-se as características de cada parte do sistema de um quadricóptero e sua importância para o bom funcionamento do VANT. O trabalho feito nesta dissertação é referente à parte do drone chamada de Controladora de Voo e ao Sensoramento que a auxilia.

Cada componente que constitui a controladora de voo aqui projetada foi descrito em detalhes, as placas escolhidas foram especificadas, assim como sua finalidade no projeto. Caracterizou-se também a conexão correta entre os dispositivos, por meio de um esboço do circuito conectado por fios e, em seguida, mostrando o projeto da PCB desenvolvida e a conexão com todos os componentes.

Ao final, o algoritmo de controle empregado para exemplificar a utilização da controladora foi exposto em um diagrama de blocos, desta vez destacando e elucidando a importância de cada parte da arquitetura da controladora para seu funcionamento. Foi feita também uma referência ao repositório [Madruga 2018], onde se verificam trechos do código de controle implementado.

Capítulo 5

Avaliação Experimental

Neste ponto do trabalho, são apresentados e analisados os resultados dos experimentos feitos no decorrer da pesquisa, a fim de verificar e validar a aplicação dos sensores escolhidos ao objetivo desejado, bem como seu funcionamento conjunto durante a operação da controladora de voo.

Começando pelos resultados das simulações dos ganhos do controlador calculados usando o método explanado na Seção 2.4, testados em duas trajetórias diferentes, seguem, nas próximas seções, experimentos feitos em cada componente de *hardware* da controladora. Por fim, capítulo é concluído com um teste de sintonia do controlador em que todos os dispositivos do projeto operam.

5.1 Simulações do Método de Sintonia Baseado em PSO

Nesta seção, discutiremos os resultados da aplicação do método de sintonia apresentado na Seção 2.4 a uma modelagem do quadricóptero no Simulink[®], adaptada de [Mahony, Kumar e Corke 2012] e [Hartman et al.]. As execuções do PSO também foram executadas no MatLab[®] e no Simulink[®], com o auxílio do mesmo modelo computacional.

O experimento consistiu de duas trajetórias diferentes, um círculo e um triângulo. Para cada uma delas, é feita uma comparação da vista superior em duas dimensões e do gráfico traçado em três dimensões. Uma análise do Erro Médio Quadrático também é apresentada ao final.

Na primeira trajetória simulada, um círculo com raio igual a $2m$, o quadricóptero começa

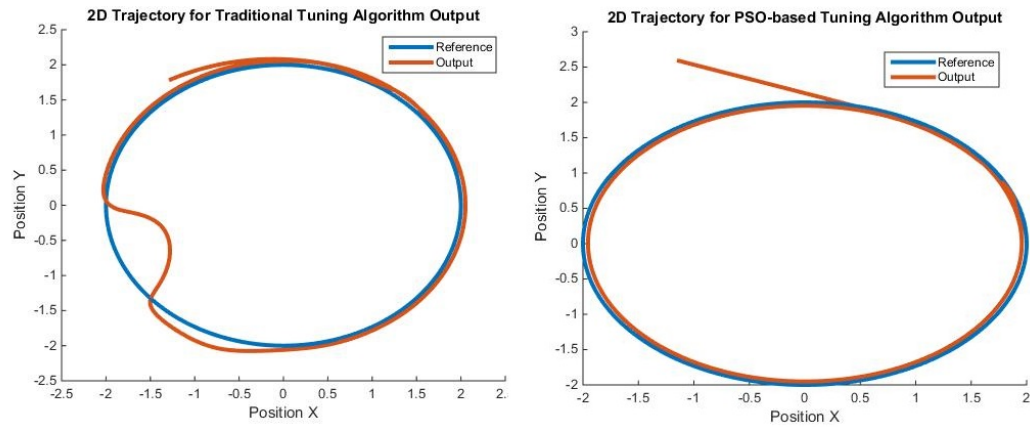


Figura 5.1: Comparação das trajetórias realizadas com a sintonia tradicional (esquerda) e com o PSO (direita).

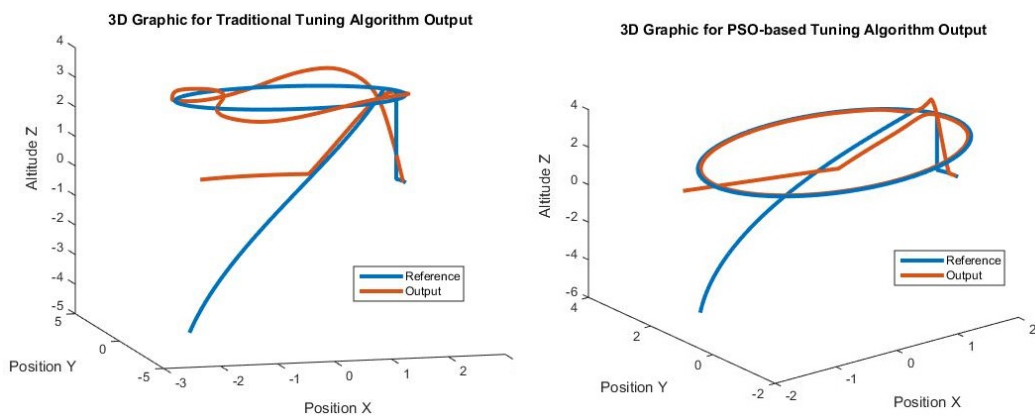


Figura 5.2: Comparação dos gráficos em 3D para a sintonia tradicional (esquerda) e com o PSO (direita).

no chão, sobe até a altura de $3m$, completa a trajetória desejada e retorna ao solo. Nas figuras analisadas a seguir, a referência é representada em azul, e a saída, em vermelho. As Figuras 5.1 e 5.2 mostram os resultados para dois métodos de sintonia, o PSO proposto, e o chamado tradicional, que se refere a uma mistura de métodos de sintonia clássicos com ajustes empíricos.

Apesar de o desvio da trajetória circular no lado esquerdo da Figura 5.1 ser bem evidente, a parte mais instável do controle com a sintonia tradicional é a altitude, que apresenta uma oscilação considerável em torno da referência de $3m$, como pode ser visto no lado esquerdo da Figura 5.2. Esses defeitos devem ser corrigidos com os ganhos obtidos com a sintonia baseada em PSO.

Com os novos ganhos aplicados ao controlador, observa-se que a altitude está muito mais estável, nota-se apenas um pequeno pico no transitório inicial, convergindo para o regime rapidamente, como mostra o lado direito da Figura 5.2. A trajetória vista em duas dimensões também foi corrigida e segue a referência completamente. O desvio visto ao final da linha vermelha no lado direito da Figura 5.1 é referente ao pouso do drone, o que pode ser verificado na Figura 5.2.

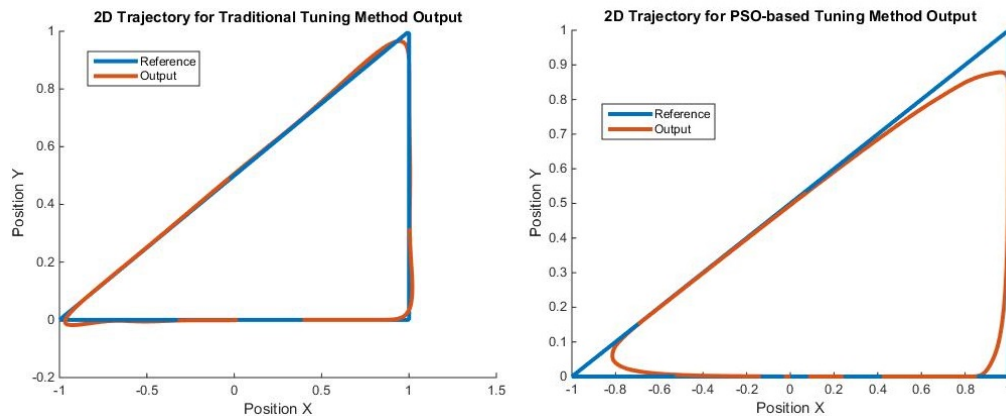


Figura 5.3: Comparação das trajetórias realizadas com a sintonia tradicional (esquerda) e com o PSO (direita).

A segunda trajetória é um triângulo retângulo isósceles com catetos de $1m$. As Figuras 5.3 e 5.4 apresentam os resultados. Observa-se que, no lado direito da Figura 5.4, referente aos ganhos do PSO, apesar de ainda haver uma oscilação, ela é da ordem de milímetros, apenas perceptível por causa da escala do gráfico.

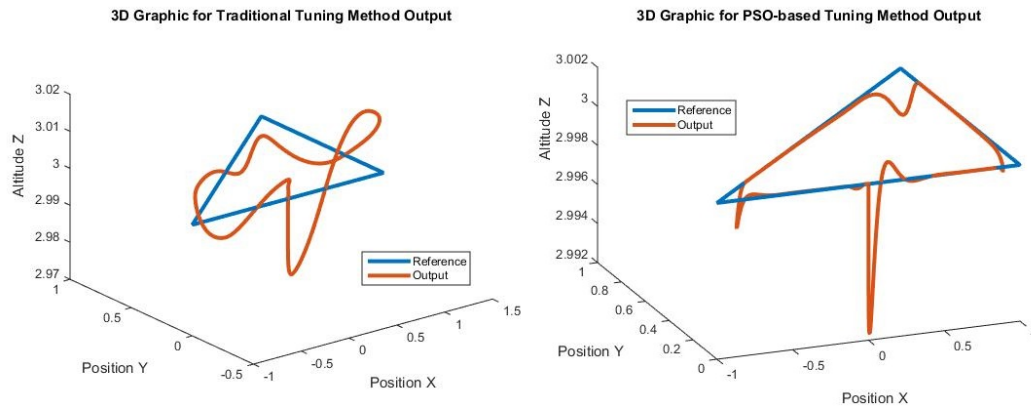


Figura 5.4: Comparação dos gráficos em 3D para a sintonia tradicional (esquerda) e com o PSO (direita).

Examinando-se a vista em duas dimensões das trajetórias triangulares, mostrada na Figura 5.3, percebe-se que a saída dos ganhos do PSO resultou em uma trajetória ligeiramente menor que a referência. No entanto, Erro Médio Quadrático ainda foi menor que na trajetória anterior, como mostra a Tabela 5.1. Os ganhos adotados em cada simulação podem ser conferidos na Tabela 5.2.

Tabela 5.1: Análise de Erro Médio Quadrático

	MSE - Círculo	MSE - Triângulo
Métodos de Sintonia Tradicionais	2.6469	0.0517
Método de Sintonia Baseado em PSO	2.3244	0.0397
Melhoria	12.18%	23.21%

5.2 Sensoriamento

Nesta seção, apresentaremos os resultados dos testes feitos com o Giroscópio MPU6050 e o Barômetro MS5611, presentes no módulo de Sensoriamento GY-86, utilizado na arquitetura da controladora desenvolvida neste projeto.

Tabela 5.2: Ganhos calculados e usados nas simulações

	Métodos de Sintonia Tradicionais	Método de Sintonia Baseado em PSO
Kp_{θ} (PD)	0.32	0.7368
Kd_{θ} (PD)	0.1	0.0030
Kp_{ϕ} (PD)	0.32	1.0815
Kd_{ϕ} (PD)	0.1	0.0099
Kp_{ϕ}	2	29.3560
Ki_{ϕ}	1.1	1.7548
Kd_{ϕ}	1.2	6.1140
Kp_{θ}	2	43.9193
Ki_{θ}	1.1	0.2127
Kd_{θ}	1.2	6.0170
Kp_{ψ}	4	0.3468
Ki_{ψ}	0.5	0.4036
Kd_{ψ}	3.5	11.4075
Kp_z	2	23.1958
Ki_z	1.1	0.0366
Kd_z	3.3	9.4616

5.2.1 Giroscópio e Acelerômetro

O giroscópio e o acelerômetro, que fazem parte do Módulo de Medição Inercial (IMU), são de extrema importância para o funcionamento do sistema como um todo. Sem eles, não há realimentação de estados, isto é, o controlador não tem como saber os dados atuais de inclinação e velocidade do quadricóptero, que são de necessidade vital para o processamento do algoritmo.

Os gráficos mostrados nas Figuras 5.5 e 5.6 traçam os resultados do teste de bancada, feito girando manualmente o drone em várias direções, e analisando a leitura recebida do IMU. O código referente a esse teste está disponível no repositório [Madruga 2018], e a biblioteca utilizada pode ser encontrada em [Martijn 2017]. Primeiramente o quadricóptero foi mantido na posição plana, virado para o norte (zero graus), por alguns momentos; logo após, foi feito um movimento de *pitch* para frente e para trás, seguido pela movimentação de *yaw*, a de *roll*, nos dois sentidos, e finalizando com o retorno à posição inicial de zero graus em todos os eixos. Essa sequência de deslocamentos é claramente representada nos dois gráficos.

As leituras feitas por sensores inerciais são conhecidas por serem muito ruidosas, principalmente quando se trata de IMUs. Para analisar os dados, aplicou-se um filtro complementar, destacado no Código ??, além de uma fusão de sensores para calcular os ângulos lidos pelo giroscópio. Ou seja, usamos as saídas do giroscópio e do acelerômetro combinadas para o cálculo dos ângulos ϕ , θ e ψ , o que implica uma maior precisão e menos erros.

Ainda assim, sempre resta um pouco de ruído no sinal, o que é esperado. Existem estudos complexos feitos apenas para tentar compensar ruídos de sensores [Hetényi, Gótzty e Blázovics 2016].

5.2.2 Barômetro

O objetivo deste experimento foi obter uma medição de altura, a partir da leitura de pressão do barômetro, e coletar dados para verificar sua precisão centimétrica. Os testes foram feitos em bancada, e limitando a altura a cerca de 30cm, justamente para aferir a exatidão das grandezas.

O Código Fonte 5.1 apresenta o algoritmo implementado para realizar a leitura de pres-

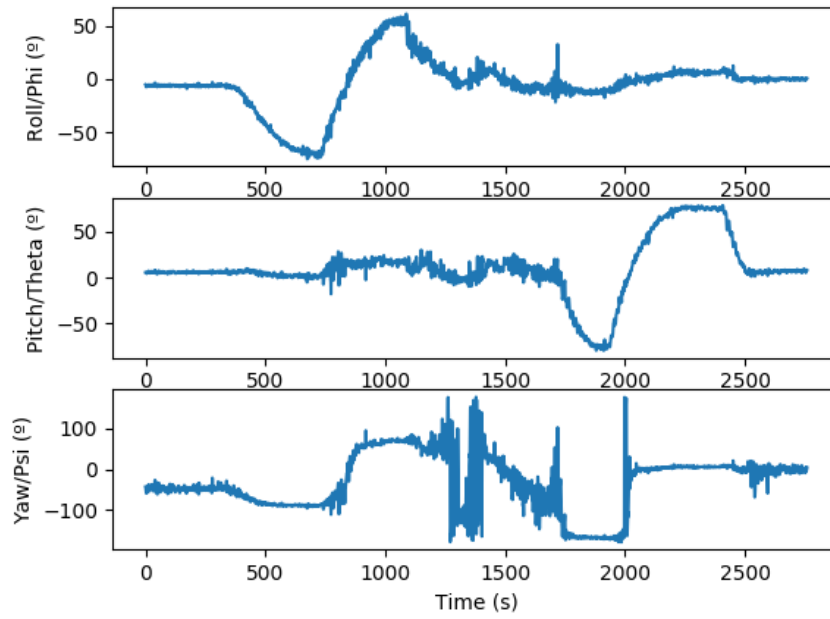


Figura 5.5: Dados de *roll*, *pitch* e *yaw* coletados durante o teste.

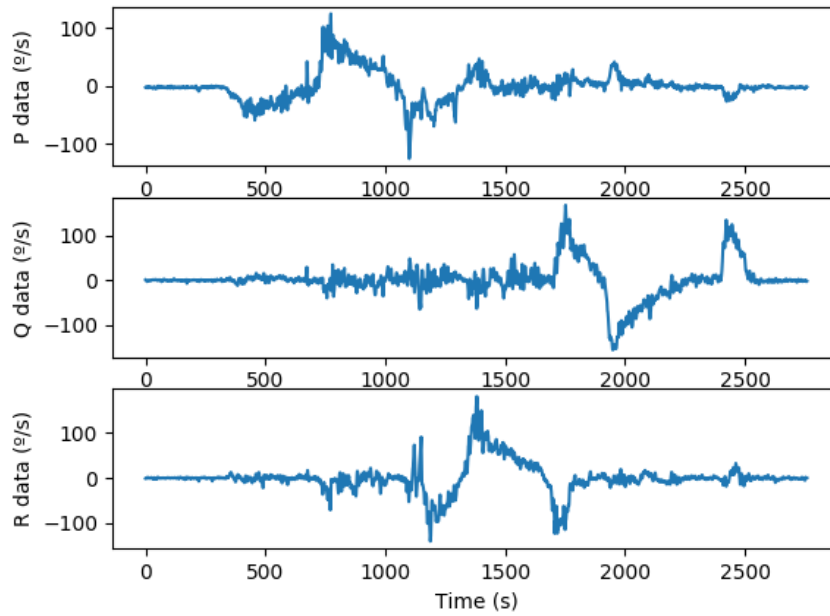


Figura 5.6: Velocidades angulares P , Q , R .

são fornecida pelo barômetro e dela derivar a altitude. Para manipular os dados recebidos, foi utilizada a biblioteca específica para o MS5611, disponível em [Staroselskii 2016]. De maneira análoga à Seção anterior, é recomendado utilizar um filtro ou uma fusão de sensores, pois todo sensor fornece leituras bastante ruidosas. Neste caso, foi empregado um Filtro Complementar.

A altura é derivada da pressão a partir da Equação 5.1, em que P é a pressão atmosférica, d é a densidade do ar, g é a aceleração da gravidade, e h é a altitude. Os resultados obtidos foram traçados no gráfico da Figura 5.7. Vemos no gráfico que a leitura é de fato muito precisa na escala centimétrica, e que o sensor tem uma boa resposta a mudanças rápidas na altitude, mesmo que pequenas.

$$P = Dgh \quad (5.1)$$

Código Fonte 5.1: Código para coletar dados do barômetro e calcular a altitude

```

1  import time
2  import navio.util
3  from navio.ms5611 import MS5611
4
5  navio.util.check_apm()
6  baro = MS5611()
7  baro.initialize()
8
9  #-----
10
11 time.sleep(2)
12 baro.refreshPressure()
13 time.sleep(0.01) # Waiting for pressure data ready 10ms
14 baro.readPressure()
15 filtered = baro.PRES
16 ref = (filtered/(1.165*9.8))
17 print("referencia: ",ref)
18 #-----
19
20 while(True):
21     baro.refreshPressure()
22     time.sleep(0.01) # Waiting for pressure data ready 10ms
23     baro.readPressure()
24
25     baro.refreshTemperature()
26     time.sleep(0.01) # Waiting for temperature data ready 10ms
27     baro.readTemperature()

```

```

28
29     baro.calculatePressureAndTemperature()
30     print ("Temperature(C): " , (baro.TEMP))
31     print ("Pressure(millibar): " , (baro.PRES))
32     #-----
33     filtered = filtered + 0.1*((baro.PRES) - filtered)
34     alt = (ref - filtered/(1.165*9.8))/3.6
35     print ("altura em cm", (alt//1))
36     #-----
37     # p=dgh, onde p eh a pressao, d eh a densidade do ar
38     # (para 25 graus = 1.1839 e para 30 graus = 1.165), g eh
39     # a gravidade, e h eh a altura
40     #-----
41
42     time.sleep(1)

```

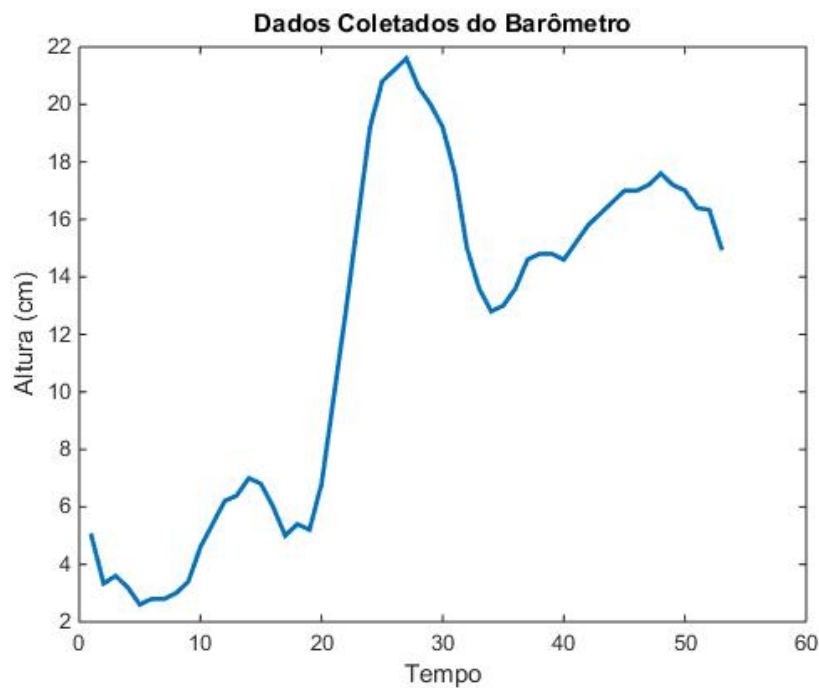


Figura 5.7: Representação gráfica da altitude derivada dos dados coletados pelo barômetro.

5.3 Módulo de Comandos PWM

A placa Adafruit PCA9685 é a responsável por enviar os comandos de velocidade dos motores para os ESCs, que os interpreta em forma de comandos de *Throttle* e os recebe em modulação por largura de pulso (PWM).

Assim, para verificar seu funcionamento, os comandos de *Throttle* em porcentagem (para melhor entendimento) foram tratados em código, convertidos para PWM e enviados aos ESCs, e deles, aos motores. A velocidade de rotação foi verificada com o auxílio de um tacômetro a laser. Os dados coletados podem ser vistos na Tabela 5.3.

Estes dados foram assim traçados em gráfico e estão representados na Figura 5.8. Esta é a curva que é aproximada linearmente pela Equação 2.21, citada no Capítulo 2. Observe-se a parte inicial do gráfico, que se mantém em zero, até 7% de *throttle*, como mostra a tabela. Isto ocorre porque o motor precisa de uma quantidade mínima de *throttle* para vencer a inércia. Essa é uma constante única que varia em cada tipo de motor e deve ser obtida experimentalmente.

Nota-se que as respostas dos motores aos comandos enviados são consistentes e formam bem a curva logarítmica exposta na Figura 5.8. A manipulação dos sinais da placa PCA9685 foi feita com o auxílio da biblioteca disponível em [DiCola 2016].

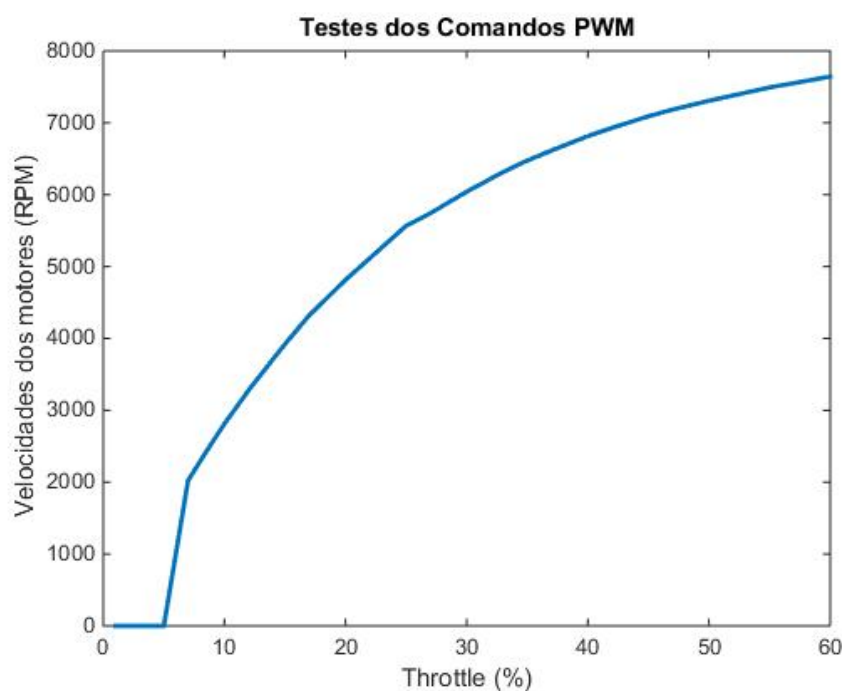


Figura 5.8: Representação gráfica dos dados coletados na Tabela 5.3.

Tabela 5.3: Dados coletados no teste de resposta aos comandos PWM

% Throttle	RPM
1	0
2	0
3	0
4	0
5	0
7	2024
8	2291
9	2555
10	2811
12	3280
15	3922
17	4321
20	4816
22	5116
25	5568
27	5742
30	6042
33	6315
35	6475
37	6615
40	6815
42	6931
45	7091
47	7185
50	7307
55	7492
60	7643

5.4 Testes de Sintonia do Controlador na Raspberry Pi

Após comprovar o funcionamento de todos os componentes que constituem o sistema, começamos a realizar os experimentos práticos para determinar os ganhos K_p , K_i e K_d adequados para as nossas circunstâncias. Códigos utilizados podem ser encontrados em [Madruça 2018].

Para efetuar tais testes, é imprescindível que todas as partes do sistema estejam conectadas, enviando e recebendo informações corretamente, caso contrário, o controlador não receberá os dados necessários para executar o algoritmo, e o processamento é interrompido. Por esse motivo os experimentos individuais explanados nas seções anteriores foram tão importantes.

Diferentemente de outras aplicações robóticas, o ajuste de sintonia do controlador de um VANT não pode ser feito sem uma plataforma de testes, por uma questão de segurança, pois se os ganhos não estiverem adequados, o drone pode voar descontroladamente e ferir alguém, ou ele pode cair e danificar os componentes. Especialmente em um caso como o estudado aqui, em que há doze valores de ganho para ajustar, pois há 4 PIDs em execução, para *roll*, *pitch*, *yaw* e altitude.

Ao final da pesquisa, projetou-se uma Gaiola de Testes para Quadricópteros, cuja patente pode ser vista no Apêndice B, que permite o livre movimento do drone em todas as direções no decorrer do teste. Sua utilização é sugerida como continuação, em forma de trabalhos futuros, na conclusão deste documento.

Para coletar os dados a seguir, imobilizou-se dois eixos (x e z) de movimentação do VANT, ao passo que, no eixo livre (y), testavam-se valores de ganhos para a estabilização do quadricóptero. Utilizou-se uma configuração de eixos em "+" para este experimento.

A Figura 5.9 mostra os comandos de saída do controlador PID para o motor chamado M2, enquanto ele tenta equilibrar o drone com os valores de ganhos testados. O processo é análogo para os outros três motores.

Na Figura 5.10, os comandos PID foram convertidos para o intervalo dos sinais PWM e enviados aos ESCs e motores através da placa PCA9685. Observe-se que eles são limitados em um valor máximo, para garantir a segurança do sistema.

A única maneira de fazer a sintonia dos ganhos do controlador corretamente é utilizando

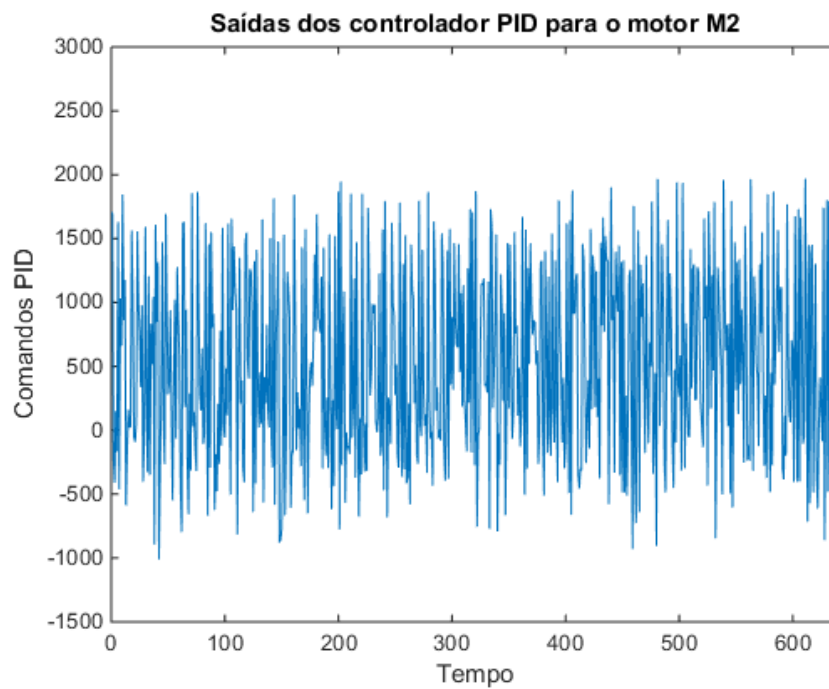


Figura 5.9: Saídas do controlador PID referentes ao motor M2 enviadas ao módulo PWM.

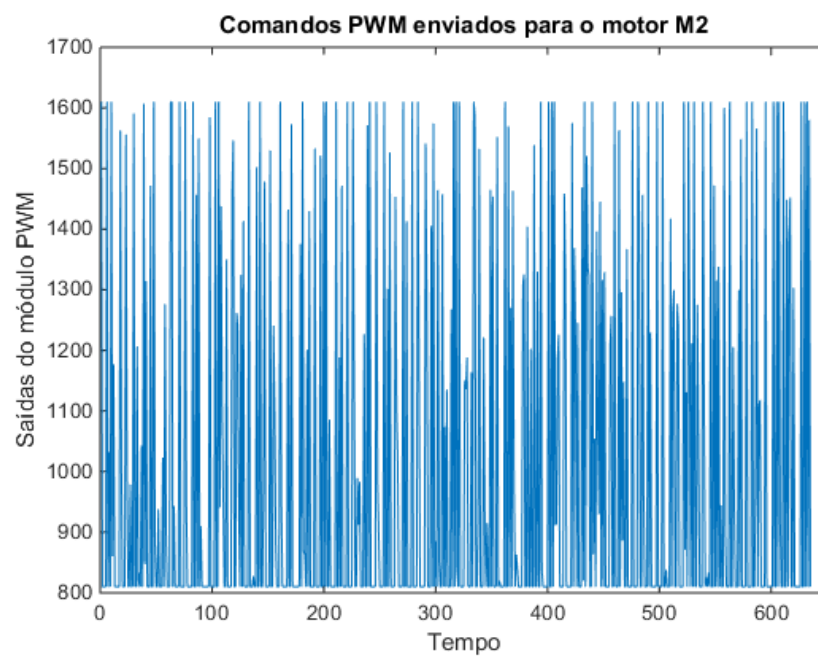


Figura 5.10: Comandos de velocidade angular enviados ao motor M2 pelo módulo PWM.

alguma plataforma de testes que forneça liberdade de movimentação ao quadricóptero, como pode ser visto em vários trabalhos [Wang et al. 2017], [Paiva et al. 2016]. O drone é um sistema altamente não linear e subatuado, a sintonia feita em cada eixo separadamente não funciona, pois as não linearidades não permitem a aplicação de superposição, por isso, sugerimos aos pesquisadores desenvolvendo seus próprios algoritmos de controle para quadricópteros a utilização da plataforma projetada no Apêndice B para a realização dos experimentos práticos.

5.5 Considerações Finais

Neste capítulo foram apresentados os resultados dos experimentos feitos no decorrer do trabalho. Primeiramente, o método para sintonia de controladores baseado em PSO foi testado em simulação, em dois tipos de trajetórias, com resultados bastante satisfatórios.

Em seguida, passamos aos experimentos relacionados aos componentes da controladora: os sensores IMU e barômetro, o módulo PWM, e o teste de sintonia do controlador com todos simultaneamente.

O tratamento dos dados lidos do IMU foi feito com fusão de sensores e uso de um filtro complementar. As respostas verificadas nos gráficos foram consistentes com as movimentações feitas no teste de bancada. Conseguimos com sucesso derivar a altura a partir da leitura de pressão do barômetro, com precisão centimétrica, mais uma vez com o auxílio de um filtro complementar.

O próximo passo foi verificar o recebimento dos comandos PWM pelos motores e seu comportamento, que foi medido com um tacômetro a laser e registrado em uma tabela. Foi traçado também um gráfico que mostrou uma curva logarítmica. Com este experimento, obtivemos a constante de *throttle* mínimo, única para cada tipo de motor e que indica o valor de *thottle* mínimo necessário para vencer a inércia e começar a girar.

No teste de sintonia do controlador PID feito com todos os componentes cumprindo suas devidas funções, a controladora comportou-se bem, de maneira estável, e os gráficos evidenciaram a saturação dos comandos enviados aos motores, que foi feita como medida de proteção.

Capítulo 6

Conclusão

Ao longo do trabalho, foi apresentado um projeto de sistema embarcado para uma controladora de voo aplicada a quadricópteros. Entre suas vantagens, podemos destacar arquitetura de *hardware* e *software* abertos, fácil acesso aos componentes, além de total autonomia do usuário para implementar seus próprios algoritmos de controle, diferentemente da imensa maioria das alternativas acessíveis disponíveis no mercado.

Cada componente envolvido foi submetido a ensaios para a verificação do funcionamento individual e conjunto de todas as partes da controladora. Como exemplo de aplicação, foi implementado um controlador PID, bastante popular para o controle de pose de quadricópteros, e os testes de sintonia experimental dos ganhos K_p , K_i , e K_d foram utilizados para constatar o bom funcionamento da arquitetura completa.

Como possibilidades de trabalhos futuros, que deem continuidade a este projeto, sugere-se primeiramente a aplicação lógica do dispositivo aqui desenvolvido: a implementação e teste de controladores mais complexos, de postura e trajetória, empregando também a plataforma de testes mostrada no Apêndice B.

Em segundo lugar, o método de sintonia experimental com a Gaiola de Testes do Apêndice B também pode ser refinado, com a execução de técnicas de *hardware-in-the-loop*, como mostrado em [Wang et al. 2017]). Outro ponto que sempre merece atenção em todo sistema embarcado, não importa o resultado obtido, é a filtragem dos ruídos nas leituras dos sensores. Podem ser empregadas técnicas de filtragem e fusão de sensores mais elaboradas, para a obtenção de resultados ainda mais precisos.

Referências Bibliográficas

- [Abbas e Sami 2017]ABBAS, N. H.; SAMI, A. R. Tuning of pid controllers for quadcopter system using hybrid memory based gravitational search algorithm – particle swarm optimization. In: . [S.l.: s.n.], 2017.
- [Anduray e Irigoyen 2017]ANDURAY, R. G. J.; IRIGOYEN, S. M. Z. Development of a fuzzy controller for liquid level by using raspberry pi and internet of things. In: *2017 IEEE Central America and Panama Student Conference (CONESCAPAN)*. [S.l.: s.n.], 2017. p. 1–5.
- [Astudillo et al. 2017]ASTUDILLO, A. et al. Altitude and attitude cascade controller for a smartphone-based quadcopter. In: *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*. [S.l.: s.n.], 2017. p. 1447–1454.
- [Ate e Abdelrahim 2018]ATE, A.; ABDELRAHIM, M. Controlling the temperature reactor based on raspberry pi system control. In: *2018 5th International Conference on Electrical and Electronic Engineering (ICEEE)*. [S.l.: s.n.], 2018. p. 215–218.
- [Basso et al. 2017]BASSO, G. F. et al. Kalman filter with dynamical setting of optimal process noise covariance. *IEEE Access*, v. 5, p. 8385–8393, 2017. ISSN 2169-3536.
- [Benitez et al. 2016]BENITEZ, W. et al. Development of an uav prototype for visual inspection of aerial electrical lines. In: *2016 Seventh Argentine Conference on Embedded Systems (CASE)*. [S.l.: s.n.], 2016. p. 7–12.
- [Bouabdallah 2007]BOUABDALLAH, S. *Design and control of quadrotors with application to autonomous flying*. Tese (Doutorado) — ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE, 2 2007.

- [Bouktir, Haddad e Chettibi 2008]BOUKTIR, Y.; HADDAD, M.; CHETTIBI, T. Trajectory planning for a quadrotor helicopter. In: *2008 16th Mediterranean Conference on Control and Automation*. [S.l.: s.n.], 2008. p. 1258–1263.
- [Bresciani 2008]BRESCIANI, T. *Modelling, Identification and Control of a Quadrotor Helicopter*. Tese (Doutorado) — Lund University, 10 2008.
- [DiCola 2016]DICOLA, T. *Python code to use the PCA9685 PWM servo/LED controller with a Raspberry Pi or BeagleBone black*. [S.l.]: GitHub repository, 2016. https://github.com/adafruit/Adafruit_Python_PCA9685.
- [Ebeid, Skriver e Jin 2017]EBEID, E.; SKRIVER, M.; JIN, J. A survey on open-source flight control platforms of unmanned aerial vehicle. In: *2017 Euromicro Conference on Digital System Design (DSD)*. [S.l.: s.n.], 2017. p. 396–402.
- [Ebeid et al. 2018]EBEID, E. et al. A survey of open-source uav flight controllers and flight simulators. *Microprocessors and Microsystems*, v. 61, p. 11 – 20, 2018. ISSN 0141-9331. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0141933118300930>>.
- [Eragamreddy e Sree 2017]ERAGAMREDDY, G.; SREE, K. R. Solar powered auto watering system for irrigation using embedded controller. In: *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*. [S.l.: s.n.], 2017. p. 2424–2428.
- [Fahlstrom e Gleason 2012]FAHLSTROM, P. G.; GLEASON, T. J. *Introduction to UAV Systems*. 4. ed. [S.l.]: John Wiley & Sons, 2012. ISBN 978-1-119-97866-4.
- [García F.R. Rubio 2012]GARCÍA F.R. RUBIO, M. O. R. Robust pid control of the quadrotor helicopter. In: *IFAC Proceedings Volumes*. [S.l.: s.n.], 2012. p. 229–234.
- [González-Sánchez et al. 2013]GONZÁLEZ-SÁNCHEZ, M. et al. Simplifying quadrotor controllers by using simplified design models. In: *52nd IEEE Conference on Decision and Control*. [S.l.: s.n.], 2013. p. 4236–4241.

- [Hartman et al.]HARTMAN, D. et al. *Parameter driven Simulink model for quadcopter simulation and control system design*. Disponível em: <<https://www.mathworks.com/matlabcentral/fileexchange/48053-quad-sim>, acesso em 23 de junho de 2018>.
- [Hetényi, Gótzty e Blázovics 2016]HETÉNYI, D.; GÓTZY, M.; BLÁZOVICS, L. Sensor fusion with enhanced kalman filter for altitude control of quadrotors. In: *2016 IEEE 11th International Symposium on Applied Computational Intelligence and Informatics (SACI)*. [S.l.: s.n.], 2016. p. 413–418.
- [Hussein e Nemah 2015]HUSSEIN, M. T.; NEMAH, M. N. Modeling and control of quadrotor systems. In: *2015 3rd RSI International Conference on Robotics and Mechatronics (ICROM)*. [S.l.: s.n.], 2015. p. 725–730.
- [Iyer et al. 2017]IYER, N. et al. Development of arm based embedded controller for mass flow controller system. In: *2017 Trends in Industrial Measurement and Automation (TIMA)*. [S.l.: s.n.], 2017. p. 1–4.
- [Li, Wang e Yi 2015]LI, X.; WANG, W.; YI, J. An embedded controller for a quadruped robot. In: *2015 IEEE International Conference on Mechatronics and Automation (ICMA)*. [S.l.: s.n.], 2015. p. 2174–2179. ISSN 2152-7431.
- [Lukmana e Nurhadi 2015]LUKMANA, M. A.; NURHADI, H. Preliminary study on unmanned aerial vehicle (uav) quadcopter using pid controller. In: *2015 International Conference on Advanced Mechatronics, Intelligent Manufacture, and Industrial Automation (ICAMIMIA)*. [S.l.: s.n.], 2015. p. 34–37.
- [Luo et al. 2014]LUO, Y. et al. Intelligent control and navigation of an indoor quad-copter. In: *2014 13th International Conference on Control Automation Robotics Vision (ICARCV)*. [S.l.: s.n.], 2014. p. 1700–1705.
- [Madruga 2018]MADRUGA, S. *Testes para validação de um sistema embarcado para controle de voo de quadricópteros*. [S.l.]: GitHub repository, 2018. <https://github.com/SarahMadruga/FlightController-Tests/>.

- [Mahony, Kumar e Corke 2012]MAHONY, R.; KUMAR, V.; CORKE, P. Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. *IEEE Robotics Automation Magazine*, v. 19, n. 3, p. 20–32, Sept 2012. ISSN 1070-9932.
- [Martijn 2017]MARTIJN. *A Python module for accessing the MPU-6050 digital accelerometer and gyroscope on a Raspberry Pi*. [S.l.]: GitHub repository, 2017. <https://github.com/Tijndagamer/mpu6050>.
- [MathWorks]MATHWORKS. *AR.Drone 2.0 Support from Embedded Coder*. Disponível em: <<https://www.mathworks.com/hardware-support/ar-drone.html>, acesso em 23 de junho de 2018>.
- [MathWorks]MATHWORKS. *PARROT Minidrones Support from Simulink*. Disponível em: <<https://www.mathworks.com/hardware-support/parrot-minidrones.html>, acesso em 23 de junho de 2018>.
- [Mazur, Wisniewski e McMillan]MAZUR, M.; WISNIEWSKI, A.; MCMILLAN, J. *Clarity from above: PwC global report on the commercial applications of drone technology*. Disponível em: <<https://www.pwc.pl/pl/pdf/carity-from-above-pwc.pdf>, acesso em 23 de junho de 2018>.
- [Mellinger 2012]MELLINGER, D. W. *Trajectory Generation and Control for Quadrotors*. Tese (Doutorado) — University of Pennsylvania, 1 2012.
- [Mhatre et al. 2015]MHATRE, V. et al. Embedded video processing and data acquisition for unmanned aerial vehicle. In: *2015 International Conference on Computers, Communications, and Systems (ICCCS)*. [S.l.: s.n.], 2015. p. 141–145.
- [Mohimani, Babaie-Zadeh e Jutten 2009]MOHIMANI, H.; BABAIE-ZADEH, M.; JUTTEN, C. A fast approach for overcomplete sparse decomposition based on smoothed ℓ^0 norm. *IEEE Transactions on Signal Processing*, v. 57, n. 1, p. 289–301, Jan 2009. ISSN 1053-587X.
- [Monterrosa et al. 2016]MONTERROSA, N. et al. Design, development and implementation of a uav flight controller based on a state machine approach using a fpga embedded

- system. In: *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*. [S.l.: s.n.], 2016. p. 1–8.
- [MRC 2017]MRC, S. *UAV Drones - Global Market Outlook (2017-2023)*. [S.l.], 2017.
- [Obayes, Al-Saedi e Mohammed 2017]OBAYES, S. A. S.; AL-SAEDI, I. R. K.; MOHAMMED, F. M. Prototype wireless controller system based on raspberry pi and arduino for engraving machine. In: *2017 UKSim-AMSS 19th International Conference on Computer Modelling Simulation (UKSim)*. [S.l.: s.n.], 2017. p. 69–74.
- [Paiva et al. 2016]PAIVA, E. et al. Modeling, simulation and implementation of a modified pid controller for stabilizing a quadcopter. In: *2016 IEEE International Conference on Automatica (ICA-ACCA)*. [S.l.: s.n.], 2016. p. 1–6.
- [Paiva et al. 2015]PAIVA, E. A. et al. Modeling and pid cascade control of a quadcopter for trajectory tracking. In: *2015 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*. [S.l.: s.n.], 2015. p. 809–815.
- [Pal, Shekhar e Sharma 2009]PAL, T.; SHEKHAR, C.; SHARMA, H. D. Design and implementation of embedded speed controller on arm for micromanufacturing applications. In: *2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies*. [S.l.: s.n.], 2009. p. 406–410.
- [Schuetz et al. 2017]SCHUETZ, M. et al. A custom lightweight uav for radar remote sensing: Concept design, properties and possible applications. In: *2017 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*. [S.l.: s.n.], 2017. p. 107–110.
- [Silva et al. 2015]SILVA, J. F. d. et al. An embedded system for aerial image processing from unmanned aerial vehicles. In: *2015 Brazilian Symposium on Computing Systems Engineering (SBESC)*. [S.l.: s.n.], 2015. p. 154–157.
- [Smyczynski, Starzec e Granosik 2017]SMYCZYNSKI, P.; STARZEC, L.; GRANOSIK, G. Autonomous drone control system for object tracking: Flexible system design with imple-

- mentation example. In: *2017 22nd International Conference on Methods and Models in Automation and Robotics (MMAR)*. [S.l.: s.n.], 2017. p. 734–738.
- [Solidakis et al. 2017]SOLIDAKIS, G. N. et al. An arduino-based subsystem for controlling uavs through gsm. In: *2017 6th International Conference on Modern Circuits and Systems Technologies (MOCAST)*. [S.l.: s.n.], 2017. p. 1–4.
- [Staroselskii 2016]STAROSELSKII, G. *Basic examples showing how to work with Navio's onboard devices using Python*. [S.l.]: GitHub repository, 2016. <https://github.com/emlid/Navio/tree/master/Python>.
- [Tijtgat et al. 2017]TIJTGAT, N. et al. Embedded real-time object detection for a uav warning system. In: *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*. [S.l.: s.n.], 2017. p. 2110–2118.
- [Vahid e Givargis 2001]VAHID, F.; GIVARGIS, T. D. *Embedded System Design: A Unified Hardware/Software Introduction*. [S.l.]: John Wiley & Sons, 2001. ISBN 978-0-471-38678-0.
- [Wang e Wang 2016]WANG, C.; WANG, W. An embedded controller for a quadrupe robot based on arm and dsp. In: *2016 IEEE International Conference on Mechatronics and Automation*. [S.l.: s.n.], 2016. p. 1090–1095.
- [Wang e Liu 2016]WANG, C.-F.; LIU, K. A novel particle swarm optimization algorithm for global optimization. *Intell. Neuroscience*, Hindawi Publishing Corp., New York, NY, United States, v. 2016, jan. 2016.
- [Wang et al. 2017]WANG, H. et al. Hardware in the loop based 6dof test platform for multi-rotor uav. In: *2017 4th International Conference on Systems and Informatics (ICSAI)*. [S.l.: s.n.], 2017. p. 1693–1697.
- [Yarlequé, Alvarez e Martínez 2017]YARLEQUÉ, M. A.; ALVAREZ, S.; MARTÍNEZ, H. J. Fmcw gpr radar mounted in a mini-uav for archaeological applications: First analytical and measurement results. In: *2017 International Conference on Electromagnetics in Advanced Applications (ICEAA)*. [S.l.: s.n.], 2017. p. 1646–1648.

[Zheng et al. 2015]ZHENG, S. et al. Design and implement a novel embedded servo controller for 6-dof motion system. In: *2015 International Conference on Fluid Power and Mechatronics (FPM)*. [S.l.: s.n.], 2015. p. 483–487.

Apêndice A

Patente *Hardware para Controladora Customizável para Quadricópteros*

Nas próximas páginas, serão anexadas cópias dos documentos relativos ao depósito da patente *Hardware para Controladora Customizável para Quadricópteros*.



Pedido nacional de Invenção, Modelo de Utilidade, Certificado de Adição de Invenção e entrada na fase nacional do PCT

Número do Processo: BR 10 2018 009992 2

Dados do Depositante (71)

Depositante 1 de 1

Nome ou Razão Social: UNIVERSIDADE FEDERAL DA PARAIBA

Tipo de Pessoa: Pessoa Jurídica

CPF/CNPJ: 24098477000110

Nacionalidade: Brasileira

Qualificação Jurídica: Instituição de Ensino e Pesquisa

Endereço: Cidade Universitária

Cidade: João Pessoa

Estado: PB

CEP: 58059-900

País: Brasil

Telefone: (83) 32167558

Fax:

Email: inova@reitoria.ufpb.br

Dados do Pedido

Natureza Patente: 10 - Patente de Invenção (PI)

Título da Invenção ou Modelo de Utilidade (54): HARDWARE PARA CONTROLADORA CUSTOMIZÁVEL PARA QUADRICÓPTEROS

Resumo: A presente invenção tem a função de receber algoritmos de controle variados para voos autônomos de quadricópteros e servir como uma plataforma para executá-los e testá-los com confiabilidade. O dito hardware para controladora customizável para quadricópteros é um dispositivo constituído por quatro partes principais: uma placa de circuito impresso (2), conectada a um módulo PWM (4), a uma placa de processamento que executa o código de controle (1), e a um módulo de sensoriamento (3). O funcionamento do dispositivo consiste em carregar um código de controle na placa (1). Os comandos do controlador serão enviados para os motores, por meio do módulo PWM (4). Os sensores (3) tratam de verificar o estado atual do quadricóptero, e assim, realimentar o controlador.

Figura a publicar: 2

Dados do Inventor (72)

Inventor 1 de 4

Nome: SARAH PONTES MADRUGA

CPF: 09535595482

Nacionalidade: Brasileira

Qualificação Física: Mestrando

Endereço: Rua Lionildo Francisco de Oliveira, 270, apto 1504, Bairro dos Estados.

Cidade: João Pessoa

Estado: PB

CEP: 58030-216

País: BRASIL

Telefone: (83) 987 852012

Fax:

Email: madrugasarah@gmail.com

Inventor 2 de 4

Nome: AUGUSTO DE HOLANDA BARRETO MARTINS TAVARES

CPF: 01262177421

Nacionalidade: Brasileira

Qualificação Física: Mestrando

Endereço: Rua Lionildo Francisco de Oliveira, 270, apto. 1504, Bairro dos Estados.

Cidade: João Pessoa

Estado: PB

CEP: 58030-216

País: BRASIL

Telefone: (83) 999 282979

Fax:

Email: augustotava@gmail.com

Inventor 3 de 4

Nome: TIAGO PEREIRA DO NASCIMENTO

CPF: 81828578568

Nacionalidade: Brasileira

Qualificação Física: Professor do ensino superior

Endereço: Avenida Mato Grosso, 741 apto. 1103B, Bairro dos Estados.

Cidade: João Pessoa

Estado: PB

CEP: 58030-080

País: BRASIL

Telefone: (83) 999 115577

Fax:

Email: tiagopn@ci.ufpb.br

Inventor 4 de 4

Nome: ALISSON VASCONCELOS DE BRITO

CPF: 02949328440

Nacionalidade: Brasileira

Qualificação Física: Professor do ensino superior

Endereço: Avenida Bahia, 900 apto. 1002A, Bairro dos Estados.

Cidade: João Pessoa

Estado: PB

CEP: 58030-130

País: BRASIL

Telefone: (83) 993 332084

Fax:

Email: alisson@ci.ufpb.br

Documentos anexados

Tipo Anexo	Nome
Comprovante de pagamento de GRU 200	00_Comprovante_Hardware.pdf
Relatório Descritivo	01_Relatorio descritivo.pdf
Reivindicação	02_Reivindicacao.pdf
Desenho	03_Desenhos.pdf
Resumo	04_Resumo.pdf

Acesso ao Patrimônio Genético

- ☒ Declaração Negativa de Acesso - Declaro que o objeto do presente pedido de patente de invenção não foi obtido em decorrência de acesso à amostra de componente do Patrimônio Genético Brasileiro, o acesso foi realizado antes de 30 de junho de 2000, ou não se aplica.

Declaração de veracidade

- ☒ Declaro, sob as penas da lei, que todas as informações acima prestadas são completas e verdadeiras.

“HARDWARE PARA CONTROLADORA CUSTOMIZÁVEL PARA QUADRICÓPTEROS”

[001] A presente invenção trata de um dispositivo para uma controladora de voo para quadricópteros autônomos, com algoritmo de controle customizável pelo usuário.

[002] Até o presente momento, as controladoras de voo disponíveis possuem algoritmos de controle relativamente simplificados e fechados para o usuário, este sendo capaz de alterar apenas os valores de constantes do controlador. Esses dispositivos presentes no mercado atualmente não permitem que o pesquisador coloque seu próprio código de controle dentro das controladoras, dificultando o estudo e aplicação de controladores mais elaborados.

[003] É sabido que o uso e aplicação de drones, principalmente aqueles do tipo quadricóptero, cresceram enormemente nos últimos anos. Sendo assim, a pesquisa nessa área torna-se cada vez mais relevante, especialmente em se tratando de algoritmos de controle para voo autônomo, que tornam o veículo mais versátil, e de algoritmos de controle mais elaborados e robustos, que dão mais estabilidade e confiança ao voo.

[004] Com o intuito de solucionar tais problemas desenvolveu-se a presente invenção, que permite que qualquer algoritmo de controle seja aplicado a um quadricóptero, por meio de um conjunto de sensores acoplados a uma placa de processamento, além de um módulo PWM (*Pulse Width Modulation*) de 16 canais que passa comandos aos motores do drone. As partes conectam-se entre si por meio de uma placa de circuito única impressa especialmente para esse sistema.

[005] A invenção poderá ser melhor compreendida através da seguinte descrição detalhada, em consonância com as figuras em anexo, onde:

[006] A **FIGURA 1** representa uma vista lateral do dispositivo;

[007] A **FIGURA 2** representa uma vista com cada peça individual em destaque, bem como suas conexões evidenciadas.

[008] A **FIGURA 3** representa uma vista da parte inferior da placa de circuito impresso.

[009] Com referência a estas **figuras**, pode-se observar a placa que recebe o algoritmo de controle e realiza o processamento **(1)**, conectada à placa de circuito impresso (PCB) **(2)**, à qual conectam-se um conjunto de sensores **(3)**, e o módulo PWM **(4)**.

[010] A placa **(1)** trata-se aqui de uma Raspberry Pi modelo 3B. Para o sensoramento **(3)**, utilizou-se um módulo inercial com 10 graus de liberdade, composto de acelerômetro, barômetro, giroscópio, magnetômetro e termômetro, de referência GY-86. Para os comandos PWM **(4)**, tem-se uma placa modelo Adafruit PCA-9685 com 16 canais.

[011] Na placa de circuito impresso **(2)**, podemos destacar a pinagem do tipo fêmea soldada para as conexões. Temos dois conectores com 6 pinos fêmea **(7)** para a conexão do módulo PWM **(4)**; um conector com 8 pinos fêmea **(8)** para a conexão do módulo de sensoramento; um conector fêmea duplo com 40 pinos fêmea **(10)** para a conexão da placa de processamento do controle **(5)**.

[012] Ainda na PCB **(2)**, observa-se um espaço **(9)** deixado para permitir a conexão do módulo PWM **(4)** com os ESCs (*Electronic Speed Controller*) dos motores. Essa conexão será feita em 4 dos 16 canais **(6)** disponíveis na placa **(4)**. Na figura 3, evidencia-se também a conexão de alimentação de 5V **(11)**.

[013] O funcionamento do dispositivo consiste em carregar um código, com o algoritmo de controle da preferência do usuário, na placa **(1)**. Os comandos do controlador serão enviados para os motores e seus respectivos ESCs, por meio do módulo PWM **(4)**. Os sensores presentes em **(3)** tratam de verificar o estado atual do quadricóptero, e assim, realimentar o controlador.

REIVINDICAÇÃO

1. “Hardware para controladora customizável para quadricópteros” caracterizado por ser constituído de uma placa de circuito impresso **(2)**, conectada a um módulo PWM que passa comandos aos motores **(4)**, a uma placa de processamento que executa o código de controle **(1)**, e a um módulo de sensoramento **(3)** com 10 graus de liberdade, contendo: acelerômetro, barômetro, giroscópio, magnetômetro e termômetro.

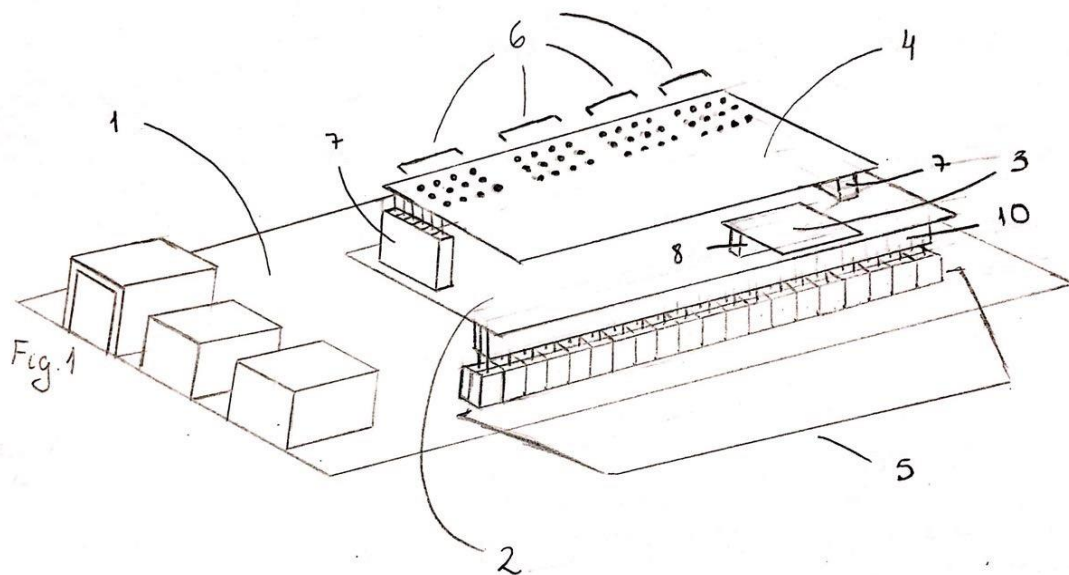


Figura 1

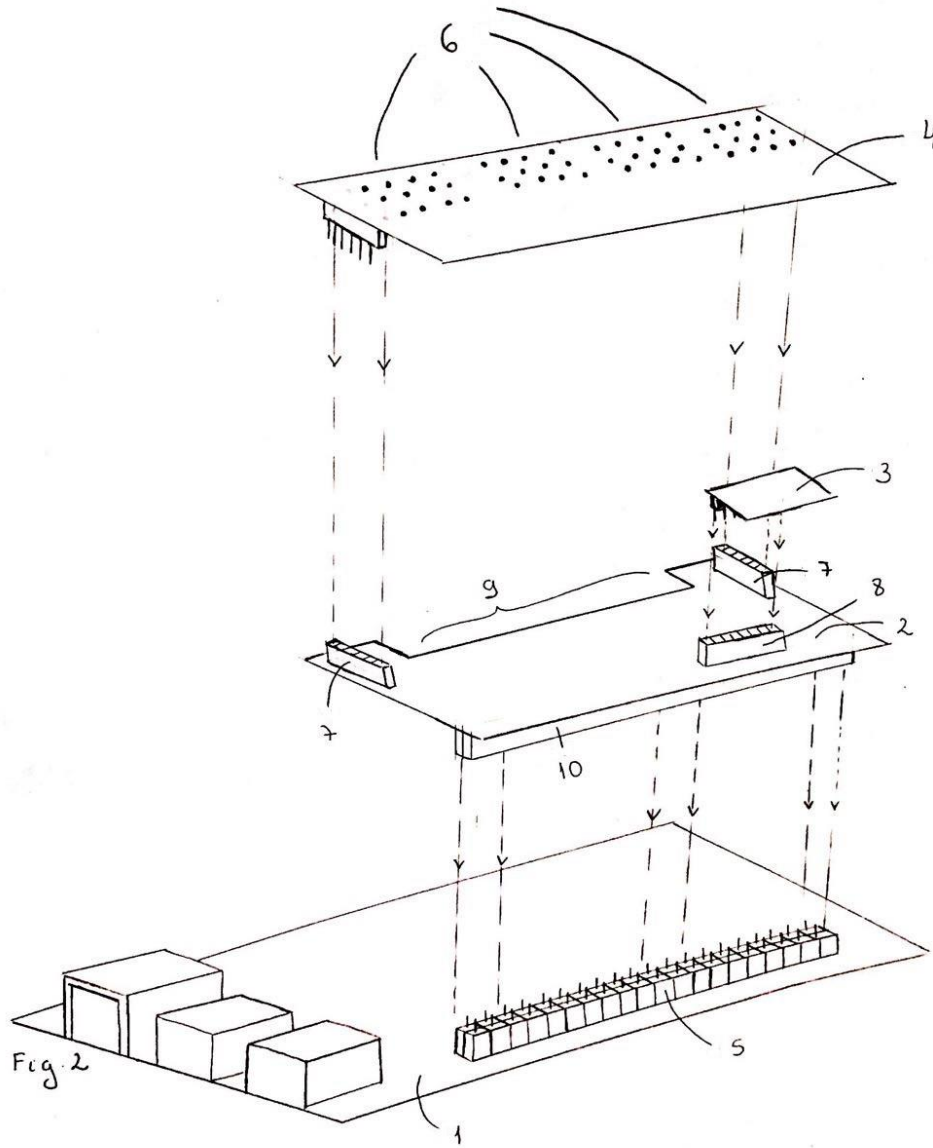


Figura 2

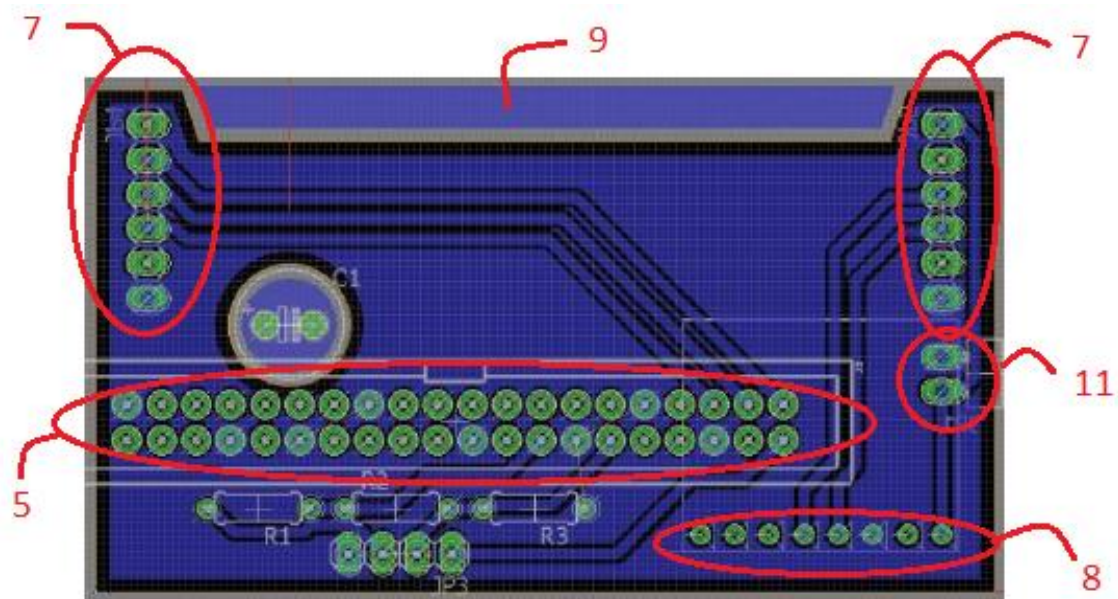


Figura 3

RESUMO

“HARDWARE PARA CONTROLADORA CUSTOMIZÁVEL PARA QUADRICÓPTEROS”

A presente invenção tem a função de receber algoritmos de controle variados para voos autônomos de quadricópteros e servir como uma plataforma para executá-los e testá-los com confiabilidade.

O dito hardware para controladora customizável para quadricópteros é um dispositivo constituído por quatro partes principais: uma placa de circuito impresso **(2)**, conectada a um módulo PWM **(4)**, a uma placa de processamento que executa o código de controle **(1)**, e a um módulo de sensoriamento **(3)**.

O funcionamento do dispositivo consiste em carregar um código de controle na placa **(1)**. Os comandos do controlador serão enviados para os motores, por meio do módulo PWM **(4)**. Os sensores **(3)** tratam de verificar o estado atual do quadricóptero, e assim, realimentar o controlador.

Apêndice B

Patente *Gaiola de Testes para Quadricópteros*

De maneira análoga ao Apêndice anterior, a seguir serão anexadas cópias dos documentos relativos ao depósito da patente *Gaiola de Testes para Quadricópteros*.



Pedido nacional de Invenção, Modelo de Utilidade, Certificado de Adição de Invenção e entrada na fase nacional do PCT

Número do Processo: BR 10 2018 010097 1

Dados do Depositante (71)

Depositante 1 de 1

Nome ou Razão Social: UNIVERSIDADE FEDERAL DA PARAIBA

Tipo de Pessoa: Pessoa Jurídica

CPF/CNPJ: 24098477000110

Nacionalidade: Brasileira

Qualificação Jurídica: Instituição de Ensino e Pesquisa

Endereço: Cidade Universitária

Cidade: João Pessoa

Estado: PB

CEP: 58059-900

País: Brasil

Telefone: (83) 32167558

Fax:

Email: inova@reitoria.ufpb.br

Natureza Patente: 10 - Patente de Invenção (PI)

Título da Invenção ou Modelo de GAIOLA DE TESTES PARA QUADRICÓPTEROS

Utilidade (54):

Resumo: A presente invenção tem como função fornecer um ambiente estável para a obtenção de resultados mais precisos em testes de quadricópteros evitando colocar cargas extras significativas sobre as hélices. A gaiola é formada por uma estrutura de alumínio (1) que fornece suporte para dois aneis (2) e (3), no centro encontra-se um sistema de sustentação (5) que fornece a base onde o quadricóptero (4) será fixado. Nas extremidades dos aneis e dos braços de sustentação temos os rolamentos (6) que permitem o quadricóptero executar os movimentos de inclinação, guinada e rolamento fornecendo assim uma estrutura confiável para a realização de testes e calibrações do controlador de voo.

Figura a publicar: 1

Dados do Inventor (72)

Inventor 1 de 6

Nome: RODRIGO PATRICIO ALVES VIEIRA

CPF: 12160138452

Nacionalidade: Brasileira

Qualificação Física: Estudante de Graduação

Endereço: Avenida Sergipe, Nº 444, apto. 201, Bairro dos Estados.

Cidade: João Pessoa

Estado: PB

CEP: 58030-190

País: BRASIL

Telefone: (83) 986 710636

Fax:

Email: rodrigo.alves@eng.ci.ufpb.br

Inventor 2 de 6

Nome: GABRIEL BARBOSA FERRAZ DE ANDRADE

CPF: 60967113326

Nacionalidade: Brasileira

Qualificação Física: Estudante de Graduação

Endereço: Rua Juiz João Navarro Filho, Nº 142, apto. 203, Jardim Oceania.

Cidade: João Pessoa

Estado: PB

CEP: 58037-250

País: BRASIL

Telefone: (83) 988 559595

Fax:

Email: gabriel.bferraz64@gmail.com

Inventor 3 de 6

Nome: SARAH PONTES MADRUGA

CPF: 09535595482

Nacionalidade: Brasileira

Qualificação Física: Mestrando

Endereço: Rua Lionildo Francisco de Oliveira, Nº 270, apto. 1504, Bairro dos Estados.

Cidade: João Pessoa

Estado: PB

CEP: 58030-216

País: BRASIL

Telefone: (83) 987 852012

Fax:

Email: madruga.sarah@gmail.com

Inventor 4 de 6

Nome: AUGUSTO DE HOLANDA BARRETO MARTINS TAVARES

CPF: 01262177421

Nacionalidade: Brasileira

Qualificação Física: Mestrando

Endereço: Rua Lionildo Francisco de Oliveira, Nº 270m apto. 1504, Bairro dos Estados.

Cidade: João Pessoa

Estado: PB

CEP: 58030-216

País: BRASIL

Telefone: (83) 999 282979

Fax:

Email: augustotava@gmail.com

Inventor 5 de 6

Nome: TIAGO PEREIRA DO NASCIMENTO

CPF: 81828578568

Nacionalidade: Brasileira

Qualificação Física: Professor do ensino profissional

Endereço: Avenida Mato Grosso, Nº 741, apto. 1103B, Bairro dos Estados.

Cidade: João Pessoa

Estado: PB

CEP: 58030-080

País: BRASIL

Telefone: (83) 999 115577

Fax:

Email: tiagopn@ci.ufpb.br

Inventor 6 de 6

Nome: ALISSON VASCONCELOS DE BRITO

CPF: 02949328440

Nacionalidade: Brasileira

Qualificação Física: Professor do ensino superior

Endereço: Avenida Bahia, Nº 900, apto. 1002A, Bairro dos Estados.

Cidade: João Pessoa

Estado: PB

CEP: 58030-130

País: BRASIL

Telefone: (83) 993 332084

Fax:

Email: alisson@ci.ufpb.br

Documentos anexados

Tipo Anexo	Nome
Comprovante de pagamento de GRU 200	00_Comprovante_Gaiola.pdf
Relatório Descritivo	01_Relatorio descritivo.pdf
Reivindicação	02_Reivindicacao.pdf
Desenho	03_Desenhos.pdf
Resumo	04_Resumo.pdf

Acesso ao Patrimônio Genético

- ☒ Declaração Negativa de Acesso - Declaro que o objeto do presente pedido de patente de invenção não foi obtido em decorrência de acesso à amostra de componente do Patrimônio Genético Brasileiro, o acesso foi realizado antes de 30 de junho de 2000, ou não se aplica.

Declaração de veracidade

- ☒ Declaro, sob as penas da lei, que todas as informações acima prestadas são completas e verdadeiras.

“GAIOLA DE TESTES PARA QUADRICÓPTEROS”

[001] A presente invenção trata de uma gaiola de testes com aplicação na área de sistemas embarcados, robótica e automação visando proporcionar um ambiente estável para a condução de testes em quadricópteros.

[002] Atualmente, os equipamentos que tem como objetivo prover um ambiente favorável para testes em quadricópteros podem influenciar no resultado final ao acrescentar forças extras sobre a estrutura do drone ou, em alguns modelos, limitam os seus movimentos.

[003] Com o intuito de solucionar tais problemas desenvolveu-se a presente invenção, onde o quadricóptero é fixado em uma base de sustentação ligada a um sistema de anéis que permitem rotações nos três eixos. Isso só se torna possível graças aos rolamentos presentes nas extremidades dos aros e dos braços que suportam a base.

[004] A invenção poderá ser melhor compreendida através da seguinte descrição detalhada, em consonância com as figuras em anexo, onde:

[005] A **FIGURA 1** representa a estrutura completa da gaiola com um drone de exemplo ao centro;

[006] A **FIGURA 2** representa a hierarquia das peças na gaiola;

[007] A **FIGURA 3** representa os braços que suportam a base;

[008] A **FIGURA 4** representa a base de sustentação do quadricóptero.

[009] Com referência a estas **figuras**, pode-se observar a gaiola que tem uma estrutura de alumínio **(1)** composta por dois anéis **(2)** e **(3)** que giram em eixos diferentes, e por um sistema de sustentação composto por uma base **(5)** com dois braços e conectados por rolamentos **(6)** que provem rotações em torno dos eixos. Vale salientar que as medidas de cada peça podem ser adaptadas conforme o tamanho do quadricóptero que será utilizado pelo usuário.

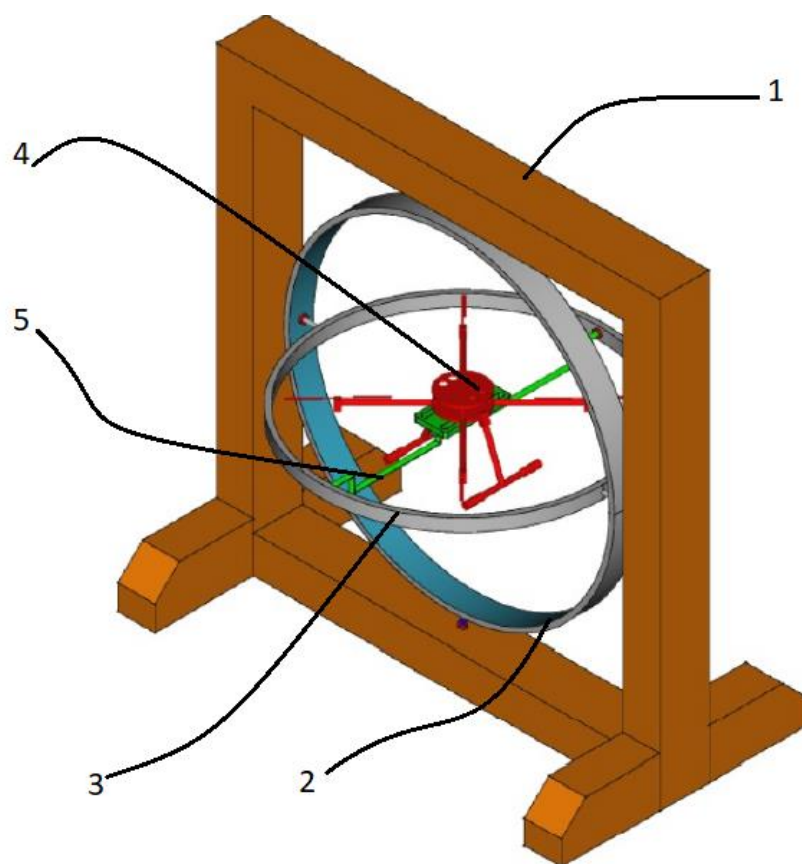
[010] A estrutura de alumínio **(1)** suporta dois aros **(2)** e **(3)** que permitem o movimento no ângulo de guinada e de inclinação. Esse movimento é possível por causa dos rolamentos **(6)** que se encontram nas extremidades.

[011] O terceiro eixo de rotação do gimbal é feito pelo sistema de sustentação **(5)**, essa montagem permite o movimento de rolamento formando o mecanismo conhecido como gimbal de três dimensões.

[012] A base da Figura 4 permite a fixação do quadricóptero a critério do usuário, podendo ser utilizado a orientação em xis ou em cruz, dependendo de como o controlador direciona a frente do quadricóptero.

REIVINDICAÇÃO

1. “Gaiola de testes para quadricópteros” caracterizado por ser constituído de uma estrutura de alumínio **(1)** que suporta os aneis **(2)** e **(3)**. No centro encontra-se um sistema de sustentação **(5)** formado por dois braços e uma base. Por fim, nos eixos de rotação encontram-se os rolamentos **(6)** que permitem o movimento radial.

**Figura 1**

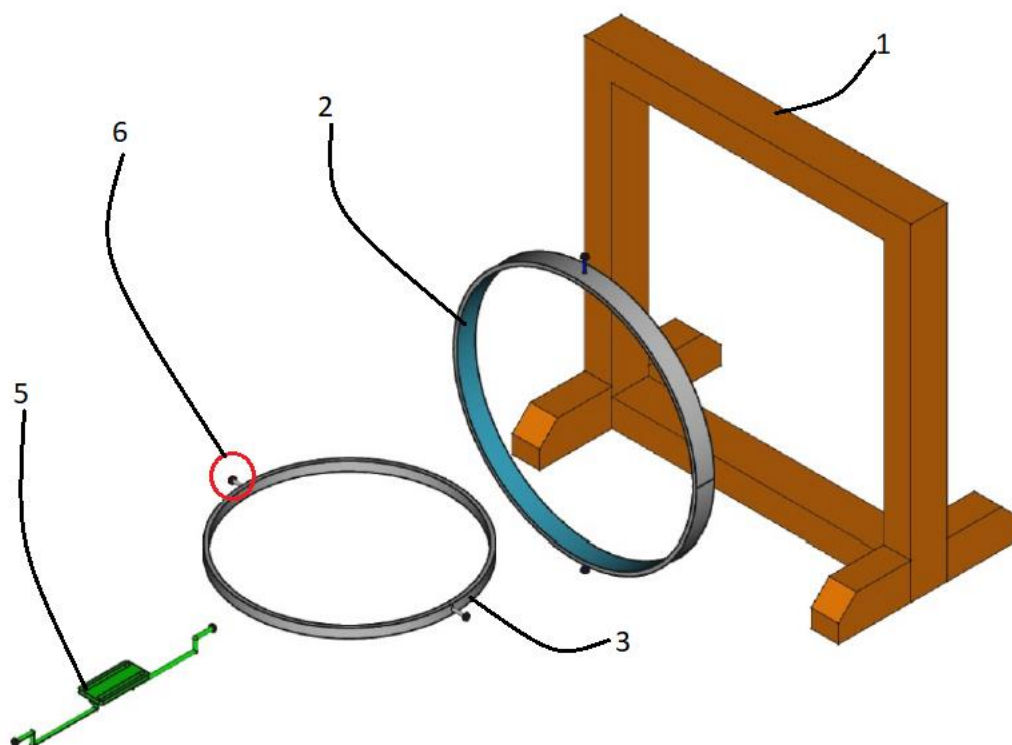


Figura 2



Figura 3

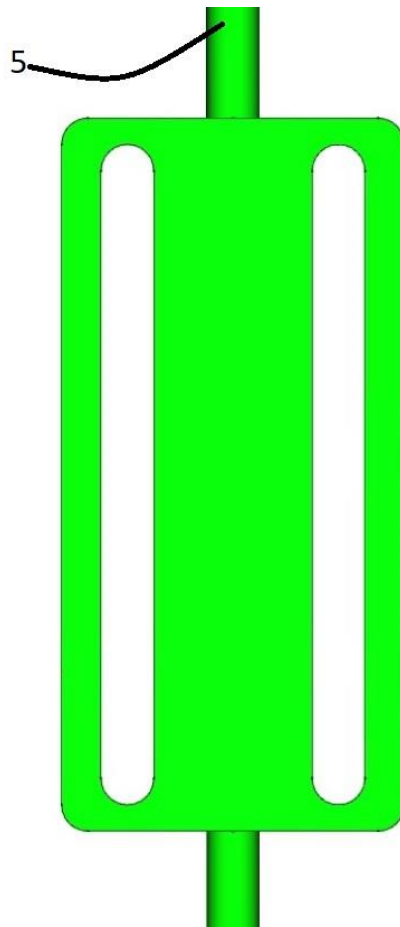


Figura 4

RESUMO

“GAIOLA DE TESTES PARA QUADRICÓPTEROS”

A presente invenção tem como função fornecer um ambiente estável para a obtenção de resultados mais precisos em testes de quadricópteros evitando colocar cargas extras significativas sobre as hélices.

A gaiola é formada por uma estrutura de alumínio **(1)** que fornece suporte para dois anéis **(2)** e **(3)**, no centro encontra-se um sistema de sustentação **(5)** que fornece a base onde o quadricóptero **(4)** será fixado.

Nas extremidades dos anéis e dos braços de sustentação temos os rolamentos **(6)** que permitem o quadricóptero executar os movimentos de inclinação, guinada e rolamento fornecendo assim uma estrutura confiável para a realização de testes e calibrações do controlador de voo.