



Universidade de Brasília – UnB  
Faculdade UnB Gama – FGA  
Engenharia Eletrônica

# Controle para automatização de um quadricóptero

Autora: Bianca Ferreira Fernandes Teixeira  
Orientador: Prof. Dr. Sébastien R. M. J. Rondineau

Brasília, DF  
2018





Bianca Ferreira Fernandes Teixeira

## **Controle para automatização de um quadricóptero**

Monografia submetida ao curso de graduação em Engenharia Eletrônica da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia Eletrônica.

Universidade de Brasília – UnB

Faculdade UnB Gama – FGA

Orientador: Prof. Dr. Sébastien R. M. J. Rondineau

Brasília, DF

2018

---

Bianca Ferreira Fernandes Teixeira

Controle para automatização de um quadricóptero/ Bianca Ferreira Fernandes  
Teixeira. – Brasília, DF, 2018-

86 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Sébastien R. M. J. Rondineau

Trabalho de Conclusão de Curso – Universidade de Brasília – UnB  
Faculdade UnB Gama – FGA , 2018.

1. controle. 2. quadricóptero. I. Prof. Dr. Sébastien R. M. J. Rondineau.  
II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Controle para  
automatização de um quadricóptero

CDU 02:141:005.6

---

Bianca Ferreira Fernandes Teixeira

## **Controle para automatização de um quadricóptero**

Monografia submetida ao curso de graduação em Engenharia Eletrônica da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia Eletrônica.

Trabalho aprovado. Brasília, DF, 24 de Agosto de 2018:

---

**Prof. Dr. Sébastien R. M. J.  
Rondineau**

---

**Prof. Dr. Henrique Marra Taira  
Menegaz**

---

**Prof. Dr. Gerardo Antonio Idrobo  
Pizo**

Brasília, DF  
2018



# Agradecimentos

Durante minha caminhada como estudante, deparei-me muitas vezes com situações que me entusiasmavam e, em outros momentos, deixavam-me desanimada. Pela esperança de um novo dia e a força daqueles que me acompanharam nesta caminhada, com carinho, dedico meus sinceros agradecimentos.

Agradeço primeiramente a Deus, por ter me dado forças nos momentos mais difíceis de minha vida, abençoando para que eu pudesse ir em frente e jamais desistir dos meus objetivos.

A minha mãe, que sempre me incentivou a estudar e que acreditou na minha capacidade .

Ao meu padrasto, que contribuiu com instrumentos e ideias.

Ao meu namorado, uma pessoa que admiro e amo muito, que sempre me incentivou a não desistir dos meus objetivos e que está sempre me ajudando a alcançá-los.

Aos amigos da faculdade , o agradecimento especial pelas experiências compartilhadas, e por terem proporcionado momentos de companheirismo que deixaram a caminhada mais leve.

Aos Mestres, em especial, ao Professor Sébastien Rondineau , pela orientação, paciência e dedicação.

E a todas as pessoas que direta ou indiretamente me ajudaram a chegar até aqui, contribuindo para que este trabalho fosse concluído.

Meu Muito obrigada



# Resumo

O presente trabalho apresenta o estudo e a análise da arquitetura de uma controladora de voo para drones, assim como, a implementação da controladora em uma plataforma para um veículo do tipo quadricóptero. Este veículo aéreo deve ser capaz de realizar auto nivelamento e garantir a estabilidade através da atuação de controle em cada um dos quatro motores, tanto quanto, empreender voos de forma autônoma a partir de trajetórias pré-definidas no software Mission Planner. Sendo assim, o objetivo principal é explicitar a implementação do projeto, a partir da placa controladora Ardupilot baseada em Arduino Mega 2560 (microcontroladores Atmega 2560 e Atmega328), detalhando o hardware, o software, o sensoriamento, assim como os seus respectivos fundamentos teóricos e ao final descrever os testes realizados e os resultados experimentais que contribuíram para o funcionamento de todo protótipo e para um controle de voo eficaz, comprovando assim o seu nível de confiabilidade.

**Palavras-chaves:** voo autônomo. quadricoptero. apm.



# Abstract

The current work presents the study and analysis of the architecture of a flight controller for drones, as well as the implementation of the controller on a platform for a quadcopter type vehicle. This aerial vehicle must be able to perform self leveling and ensure stability through control actuation in each of the four engines, as well as make flights autonomously from predefined trajectories in Mission Planner software. Therefore, the main objective is to make explicit the implementation of the project, from the Ardupilot controller board based on Arduino Mega 2560 (Atmega 2560 and Atmega328 microcontrollers), detailing the hardware, the software, the sensing, as well as their respective theoretical and at the end, describe the simulations, the tests performed and the experimental results that contributed to the operation of each prototype and to an effective flight control, thus proving its level of reliability.

**Key-words:** autonomous flight. quadcopter. apm.



# Lista de ilustrações

Figura 1 – Drone Mavic Pro da marca DJI . . . . .	20
Figura 2 – Primeiro Quadrirotor, Bréguet-Richet Quad-Rotor Helicopter, 1907 . . . . .	23
Figura 3 – Aquisição de dados pelos sensores (azul) e saída de dados nos atuadores (vermelho) . . . . .	25
Figura 4 – Interface do Software Mission Planner . . . . .	28
Figura 5 – Dados de voo apresentados no Software Mission Planner . . . . .	28
Figura 6 – IDE do Arduino . . . . .	29
Figura 7 – Relação dos sinais PWM e a potência do motor . . . . .	31
Figura 8 – Satélites no espaço . . . . .	32
Figura 9 – Círculo formado pelo satélite . . . . .	32
Figura 10 – Encontro das circunferências dos satélites que irá definir a posição do receptor GPS . . . . .	33
Figura 11 – Modelo real tridimensional do GPS . . . . .	33
Figura 12 – Ligação entre a APM e o módulo GPS Ublox . . . . .	34
Figura 13 – Módulo Ublox na haste . . . . .	35
Figura 14 – À esquerda: barômetro embarcado na placa APM, à direita: barômetro digital . . . . .	36
Figura 15 – Motor Brushless com rotor externo . . . . .	36
Figura 16 – Hélices DJI 9443 E Motores 2212 DJI . . . . .	38
Figura 17 – Rádio Controle e receptor FS-i6 . . . . .	39
Figura 18 – Frame F450 . . . . .	39
Figura 19 – Bateria de lítio da marca Tattu utilizada no projeto . . . . .	40
Figura 20 – Graus de liberdade do quadricóptero . . . . .	41
Figura 21 – Princípio de Funcionamento . . . . .	42
Figura 22 – Movimentos verticais . . . . .	42
Figura 23 – Movimento Yaw . . . . .	43
Figura 24 – Movimento Pitch . . . . .	43
Figura 25 – Movimento Roll . . . . .	44
Figura 26 – Diagrama de Blocos em malha fechada . . . . .	46
Figura 27 – Fórmula que descreve o controlador PID . . . . .	46
Figura 28 – Parâmetros do modo AltHold . . . . .	47
Figura 29 – Controle PID do modo AltHold . . . . .	48
Figura 30 – Parâmetros do Loiter . . . . .	49
Figura 31 – Waypoints . . . . .	49
Figura 32 – Parâmetros do modo Auto . . . . .	50
Figura 33 – Gráfico de desempenho do AltHold . . . . .	51

Figura 34 – Gráfico de desempenho do Loiter . . . . .	51
Figura 35 – Características complementares dos sensores GPS e acelerômetro. . . . .	52
Figura 36 – Esquemático do processo de fusão dos sensores GPS e acelerômetro. . . . .	54
Figura 37 – Etapas da metodologia e suas características . . . . .	55
Figura 38 – Esquemático do teste de empuxo . . . . .	59
Figura 39 – Teste de empuxo . . . . .	60
Figura 40 – Controladores de velocidade soldados à placa de distribuição . . . . .	61
Figura 41 – Saída BEC do ESC . . . . .	62
Figura 42 – Braços e motores parafusados à estrutura . . . . .	62
Figura 43 – Esquemático de montagem do protótipo . . . . .	63
Figura 44 – Vista lateral do veículo aéreo . . . . .	64
Figura 45 – Vista superior do veículo aéreo . . . . .	64
Figura 46 – Drone completo após montagem . . . . .	64
Figura 47 – Calibração do Rádio com limites dos sticks e chaves em vermelho. . . . .	65
Figura 48 – Rádio controle e seus elementos . . . . .	66
Figura 49 – Aquisição dos limites do rádio controle . . . . .	67
Figura 50 – Algoritmo do Trim responsável pela mudança dos sinais PWM . . . . .	67
Figura 51 – Calibração da Bússola . . . . .	68
Figura 52 – Arquitetura dos modos Stabilize e AltHold . . . . .	70
Figura 53 – Arquitetura dos modos Loiter e Auto . . . . .	73
Figura 54 – Gráfico de Posição no Eixo X . . . . .	74
Figura 55 – Gráfico de Erro Relativo por Tempo das medidas da posição em X . . . . .	75
Figura 56 – Gráfico de Posição no Eixo Y . . . . .	75
Figura 57 – Gráfico de Erro Relativo por Tempo das medidas da posição em Y . . . . .	76
Figura 58 – Gráfico de Altitude . . . . .	76
Figura 59 – Gráfico de Erro Relativo por Tempo das medidas de Altitude . . . . .	77
Figura 60 – Gráfico de Velocidade no Eixo X . . . . .	77
Figura 61 – Gráfico de Erro Relativo por Tempo das medidas de Velocidade em X . . . . .	78
Figura 62 – Gráfico de Velocidade no Eixo Y . . . . .	78
Figura 63 – Gráfico de Erro Relativo por Tempo das medidas de Velocidade em Y . . . . .	79

# Lista de tabelas

Tabela 1 – Cálculo da aplicação comercial dos drones por setor de economia. . . .	19
Tabela 2 – Peso Estimado do Drone . . . . .	37
Tabela 3 – Dados comparativos entre os rádio-controles mais comumente utilizados	38
Tabela 4 – Resultados dos testes de empuxo . . . . .	60



# Lista de abreviaturas e siglas

VANT	Veículo Aéreo Não Tripulado
PID	Proporcional, Integral e Derivativo
GPS	Global Positioning System
ESC	Electronic Speed Controller
PWM	Pulse Width Modulation
rpm	Rotações por minuto
CI	Circuito Integrado
DMP	Digital Motion Processor
SCL	Serial Clock
SDA	Serial Data
IDE	Integrated Drive Electronics
ESC	Electronic Speed Controller



# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>19</b>
<b>1.1</b>	<b>Objetivo Geral</b>	<b>20</b>
<b>1.2</b>	<b>Objetivos Específicos</b>	<b>20</b>
<b>2</b>	<b>QUADRICÓPTEROS</b>	<b>23</b>
<b>2.1</b>	<b>Como funciona o voo autônomo</b>	<b>24</b>
<b>2.2</b>	<b>Placa Controladora de Voo</b>	<b>25</b>
2.2.1	ArduPilot	26
<b>2.3</b>	<b>Plataformas de Desenvolvimento</b>	<b>27</b>
2.3.1	Mission Planner	27
2.3.1.1	Programação em Arduino	29
<b>2.4</b>	<b>Sensores Essenciais para o Voo autônomo</b>	<b>30</b>
2.4.1	Unidade Inercial	30
2.4.1.1	Acelerômetro	30
2.4.1.2	Giroscópio	30
2.4.2	Controlador Eletrônico de Velocidade	31
2.4.3	GPS (Global Positioning System)	31
2.4.3.1	Funcionamento	31
2.4.3.2	Aplicação em um veículo aéreo	34
2.4.4	Bússola	34
2.4.5	Barômetro	35
2.4.6	Motores e Hélices	36
2.4.7	Rádio controle e receptor de comandos	38
2.4.8	Estrutura	39
2.4.9	Alimentação	39
<b>3</b>	<b>TEORIA DE OPERAÇÃO DO QUADRICÓPTERO</b>	<b>41</b>
<b>4</b>	<b>MODOS DE VOO</b>	<b>45</b>
<b>4.1</b>	<b>Análise de desempenho a partir de logs</b>	<b>50</b>
<b>4.2</b>	<b>Aquisição da Velocidade e Posição</b>	<b>51</b>
<b>5</b>	<b>METODOLOGIA</b>	<b>55</b>
<b>5.1</b>	<b>Levantamento de variáveis e requisitos</b>	<b>55</b>
<b>5.2</b>	<b>Tomada de Decisões</b>	<b>56</b>
<b>5.3</b>	<b>Desenvolvimento do Protótipo</b>	<b>56</b>

<b>5.4</b>	<b>Testes dos Subistemas</b>	<b>56</b>
<b>5.5</b>	<b>Validação do projeto</b>	<b>57</b>
<b>6</b>	<b>RESULTADOS</b>	<b>59</b>
<b>6.1</b>	<b>Validação dos Motores</b>	<b>59</b>
6.1.1	Teste de Empuxo	59
<b>6.2</b>	<b>Desenvolvimento do Protótipo</b>	<b>61</b>
<b>6.3</b>	<b>Calibração dos Sensores e Dispositivos</b>	<b>65</b>
6.3.1	Calibração do rádio controle	65
6.3.1.1	Trim	66
6.3.1.2	Calibração da Unidade Inercial	68
6.3.1.3	Calibração da Bússola (Compass)	68
6.3.1.4	Calibração dos ESC's	69
6.3.2	Modos de Voo	69
6.3.2.1	Independentes de GPS	69
6.3.2.2	Dependentes de GPS	72
<b>6.4</b>	<b>Análise de Desempenho a partir de logs de dados</b>	<b>74</b>
6.4.1	Posição no Eixo X	74
6.4.2	Posição no Eixo Y	75
6.4.3	Altitude	76
6.4.4	Velocidade no eixo X	77
6.4.5	Velocidade no eixo Y	78
<b>7</b>	<b>CONSIDERAÇÕES FINAIS</b>	<b>81</b>
<b>8</b>	<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>	<b>83</b>

# 1 Introdução

A utilização de veículos aéreos não tripulados (VANT's) está cada vez mais presente na sociedade, estes veículos são “mini aeronaves” capazes de sustentar voo sem a presença de uma pessoa a bordo e, para alcançar este fim, todo o controle de voo deve ser feito por um microcontrolador já instalado no VANT, que recebe comandos pré-programados ou de uma base remota e, com auxílio de diversos sensores, como giroscópio, acelerômetro, barômetro e gps, é capaz de manter-se em sua trajetória de voo de acordo com as condições do meio em que está presente (GARRET, 2015).

Devido a sua gama de aplicações, o desenvolvimento deste equipamento tornou-se uma ótima alternativa para soluções que necessitam de mobilidade em diversas áreas, como pode ser observado na Tabela 1, que apresenta o valor aplicado em cada setor da economia para utilização de drones.

Tabela 1 – Cálculo da aplicação comercial dos drones por setor de economia.

Área da Economia	Valor (bilhões de dólares)
Infraestrutura	Noruega
Transportes	Austrália
Seguros	EUA
Mídia e Entretenimento	Holanda
Telecomunicações	Alemanha
Agricultura	32,4
Segurança	10,5
Mineração	4,3
Total	127,3

De acordo com Vieira (2011), os VANT's podem ser divididos em duas categorias, os de asas fixas e os de asas rotativas, possuindo estes últimos algumas características que apresentam uma clara vantagem em comparação a outros tipos de veículos aéreos quando leva-se em conta a decolagem e a aterrissagem em espaços limitados, a habilidade de se manter em suspensão e imóvel numa determinada localização por longos períodos e a capacidade de pairar e voar a baixas velocidades.

Dentre os tipos de VANT's de asas rotativas, há ainda um subconjunto que merece destaque, que é o objeto de estudo deste trabalho, que são os quadrirotores. Um quadrirotor é um tipo de veículo cuja propulsão é realizada por quatro motores, como pode ser visto na Figura 1. As quatro hélices devem ser posicionadas de forma horizontal e o sentido de rotação dos rotores deve ser de tal forma que os motores adjacentes devem girar em sentido contrário ao de referência, para compensar o momento angular dos pro-

pulsos. Por ser um sistema dinâmico, ao se alterar a velocidade dos motores, a posição, inclinação e velocidade do quadricóptero também serão alteradas.



Figura 1 – Drone Mavic Pro da marca DJI

Devido às múltiplas situações que um drone pode atravessar, o sistema de controle embarcado necessita cada vez mais de autonomia e aperfeiçoamento, exigindo cada vez menos intervenção humana nos seus voos, para que dependa única e exclusivamente dos seus próprios sensores para o desempenho de missões (ROBERTS, STIRLING, ZUFFE-REY E FLOREANO, 2007).

Então, por causa de suas inúmeras funcionalidades, buscando eficiência e qualidade de voo e deste modo ser capaz de concluir com segurança o objetivo para qual foi designado, cabe à controladora de voo, ou flight controller, receber o plano de voo e calcular como será a trajetória do equipamento, utilizando técnicas de controle baseadas em informações obtidas dos sensores e comandos da controladora, que funciona como unidade de processamento de bordo, a partir de trajetórias pré-definidas (BARATO, 2014).

## 1.1 Objetivo Geral

Este trabalho tem como objetivo o desenvolvimento de um veículo aéreo não tripulado que tenha capacidade de realizar voos autônomos a partir de coordenadas geográficas pré-programadas na controladora de voo Ardupilot. Será realizada a análise e descrição da arquitetura responsável pelo voo autônomo, a implementação do projeto em plataforma com quatro motores, a dinâmica e a comunicação do veículo aéreo.

## 1.2 Objetivos Específicos

Esse trabalho possui como objetivos específicos:

- A análise e descrição da arquitetura de controle em linguagem C++, utilizada pela controladora de voo ArduPilot Mega 2560, capaz de realizar voos autônomos baseando-se em trajetórias pré-programadas pelo software Mission Planner

- A implementação do sistema autônomo na plataforma de um quadricóptero
- A calibração e configuração de sensores, como: unidade de medição inercial, barômetro, GPS e bússola.



## 2 Quadricópteros

Os quadrirotores, ou quadricópteros, recentes são veículos de dimensões reduzidas baseados em uma estrutura cruzada que possui em cada um dos seus quatro cantos um rotor fixo e no seu centro todos os dispositivos responsáveis por seu controle. Apesar de possuir tamanho e peso reduzidos atualmente, o primeiro quadrirotor a ser desenvolvido possuía 578 quilos e só era capaz de levantar-se a 1,5 metro, como pode ser visto na Figura 2. Segundo LEISHMAN (2002) este quadrirotor foi desenvolvido em 1907 pelos irmãos Breguet, com o nome Bréguet-Richet Quad-Rotor Helicopter, e a ideia inicial era transportar no centro dois pilotos que controlassem os quatro rotores que tinham potência fornecida por motores a gasolina e transmissão de correia.

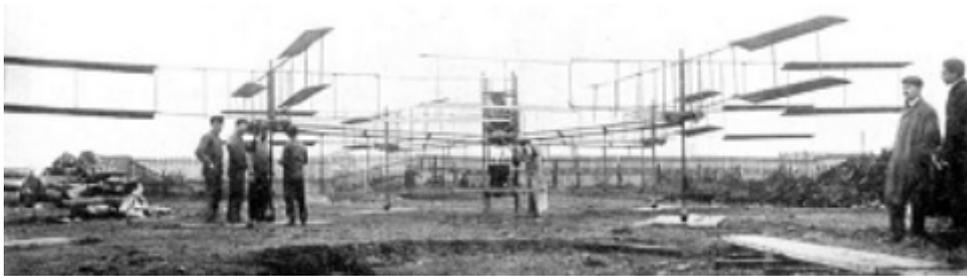


Figura 2 – Primeiro Quadrirotor, Bréguet-Richet Quad-Rotor Helicopter, 1907

Apesar de possuir voo instável e baixo, esse veículo contribuiu grandemente para o projeto dos quadricópteros, pois introduziu o princípio dos pares de rotores dos quadrirotores girarem em sentidos opostos (LEISHMAN, 2002).

Com a crescente evolução tecnológica houve a miniaturização de componentes eletrônicos e, hoje em dia, os quadrirotores possuem diversos sensores e sistemas de controle para obter sua estabilização, possuindo a grande vantagem de serem controlados por controle remoto, semi-autonomamente ou completamente autônomo (COSTA, 2008).

Um drone autônomo pode ser muito utilizado para tarefas que precisam cobrir longas distâncias em pouco tempo, que cubram locais de difícil acesso ou que seriam economicamente inviáveis se fossem realizadas por aeronaves comuns (Drones Brasil, 2017).

Então, para realizar o controle autônomo do veículo, é necessária a implementação de um programa de controle para voos autônomos, esta realização requer uma placa controladora de voo que possua um microcontrolador que seja capaz de realizar a leitura dos sensores em tempo real, assim como, efetivar a atuação nos respectivos motores.

## 2.1 Como funciona o voo autônomo

Veículos aéreos autônomos já são uma realidade hoje, eles têm como função realizar um piloto automático que efetua, auxilia e reduz as tarefas realizadas pelo piloto real, melhorando assim a eficiência e segurança dos procedimentos de voo pois diminui a possibilidade de erros humanos (CAMPELLO, 2018).

Para ANAC (2018), o piloto automático é um sistema que comanda a aeronave automaticamente, até estabilizá-la em uma condição de voo pré-estabelecida pelo computador ou ainda, segundo a mesma fonte, o piloto automático é um mecanismo de comando para manter uma aeronave em voo nivelado e sob ajuste de ruma, controlando o avião automaticamente em relação aos seus eixos de movimento angular: longitudinal, lateral e direcional (eixos  $x, y$  e  $z$ ).

Sendo assim, qualquer mecanismo que possua autonomia deve ter em seu sistema alguns sensores responsáveis por realizar leituras específicas para o qual foi programado, seja realizar leituras de movimentos, atmosféricas, de velocidade ou reconhecer o meio a sua volta.

Segundo Campello (2018), esses sensores são responsáveis pela aquisição de informações em forma de dados e as transferem para um sistema de controle, que tem o papel de interpretação de dados, realização de cálculos e aplicação de lógicas, correlacionando com um banco de dados preestabelecidos, chamadas de técnicas de controle que irão fazer as devidas alterações, correções e ações, a fim de cumprir o plano de voo predeterminado e enviar o resultado para os atuadores.

Os sistemas de controle são aptos para processar algoritmos avançados, todos os dados são transportados para a placa controladora por meio de uma porta de entrada, onde os sensores (unidade inercial, barômetro, bússola e GPS) estão conectados. Uma vez que os dados são colhidos e processados, o sistema de controle realiza por uma porta de saída a comunicação para um dispositivo periférico, seja ele um atuador, como os controladores de velocidade, ou sensores, que por sua vez alimentam o sistema novamente com novos dados.

A figura 3 apresenta os sensores que fornecem os dados de entrada, na cor azul, ligados à placa controladora e os atuadores, na cor vermelha, controladores de velocidade (ESC's) e motores, que recebem os dados de saída e atuam no veículo aéreo. A controladora de voo a partir dos dados dos sensores terá um sistema de percepção apurado e poderá atuar nos motores de acordo com a rota pré-definida em seu sistema.

No voo autônomo, o sistema de controle é carregado antecipadamente com uma rota (pelo software Mission Planner que será apresentado posteriormente) e essa trajetória trará referências imprescindíveis que vão funcionar como uma matriz, onde todas os dados colhidos irão passar por um processo para se adequar ao plano inicial.

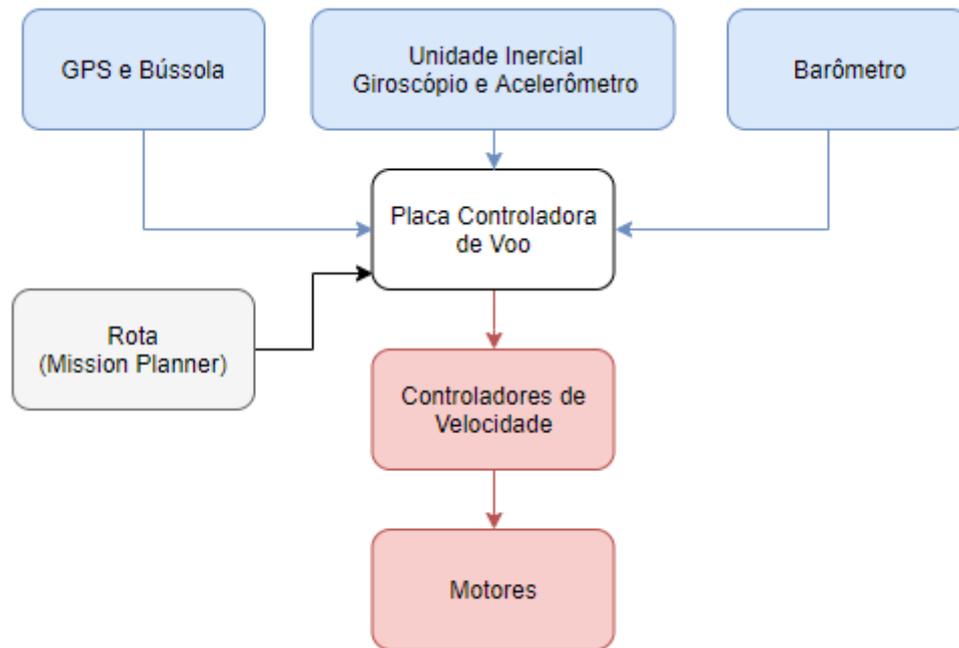


Figura 3 – Aquisição de dados pelos sensores (azul) e saída de dados nos atuadores (vermelho)

Veja um exemplo: Um drone recebe um vento forte, o qual faz com que a rota do veículo seja deslocada, dessa forma, os sensores captam essa informação advinda de equipamentos de GPS e leitores atmosféricos, transmitem para o sistema de controle que calcula com eficiência as alterações sofridas e possíveis correções aplicando assim de forma precisa as devidas ações, visando cumprir a ordem especificada em seu programa, que é de se manter uma trajetória específica (CAMPELLO, 2018).

Responsável por produzir movimentos a partir de uma requisição do sistema, os atuadores são elementos que têm como função realizar uma ação para modificar ou corrigir a atitude do equipamento, visando cumprir a proposta inicial dos dados inseridos na placa controladora.

Portanto, um sistema autônomo é composto essencialmente pela ligação e comunicação dos elementos apresentados e é capaz de compreender a própria situação de acordo com que foi programado, fazendo com que tenha ações e reações baseadas na sua rota pré-definida.

## 2.2 Placa Controladora de Voo

Existem diversos modelos de placas no mercado, que são projetadas já com diversos componentes integrados como os giroscópios, acelerômetros, barômetros e GPS, além de circuitos de comunicação, antenas e transceptores.

A controladora de voo é o componente principal de um multirrotor, ele funciona

como o “cérebro” do veículo aéreo, possui um sistema de percepção (DIAS, 2017) responsável pela leitura e processamento dos sinais de entrada (leitura de sensores), interpretação dos comandos (seja os advindos de uma rota pré-definida ou de um controle remoto) e gera as saídas adequadas para cada fase do voo, mantendo a estabilidade do drone.

Atualmente, as controladoras de voo mais utilizadas são:

- **KK** - Essa placa possui um microcontrolador Atmel Mega644PA de 8 bits, firmware próprio com interface gráfica que auxilia na configuração do drone de acordo com o número de braços, ajusta os ganhos PI, calibra os controladores de velocidade e possuem acelerômetro e giroscópio, mas não realiza voo autônomo. Custa em média 200 reais.
- **Naza** - Possui um microcontrolador de 8 bits, firmware próprio com interface gráfica para configuração, acelerômetro, giroscópio e barômetro embutidos, e como opcional um GPS que é capaz de deixar o drone autônomo. Custa em médio 589 reais.
- **Ardupilot** - é uma controladora de voo de código aberto com capacidade de voo autônomo baseada na plataforma Arduino, possui firmware open-source, processador de 8 bits e sensores para navegação embutidos. Custa, com GPS e telemetria, 360 reais.
- **Pixhawk** - é uma controladora de voo que possui os projetos de hardware e códigos abertos, possui função de voo autônomo, processador de 32 bits e os mesmos sensores que a Ardupilot. Custa R\$ 1380 com kit de telemetria.

Nesse sentido, por causa de sua extensa documentação disponível, o foco deste trabalho é a análise e compreensão do algoritmo para voos autônomos da controladora de voo ArduPilot.

### 2.2.1 ArduPilot

A ArduPilot Mega (APM) é uma controladora de voo de código aberto com capacidade de voo autônomo baseada na plataforma Arduino (DOCTOR DRONE, 2015). Ela é um pequeno computador capaz de prover estabilização autônoma, navegação por waypoints (coordenadas GPS) e telemetria (envia dados úteis para a estação em terra, como status da bateria).

Características:

- Processador Atmega 2560 e um co-processador Atmega328
- MPU6000 - Unidade Inercial com Acelerômetro e Giroscópio

- Sensor de pressão barométrica Measurement Specialties MS5611-01BA03
- Dimensões: 67x40x10mm
- Peso: 17g

Funções Essenciais:

- Calibração dos controladores eletrônicos de velocidade
- Calibração dos sensores acelerômetro e giroscópio
- Aquisição de dados dos sensores acelerômetro e giroscópio
- Recepção de comandos do rádio-controle
- Recepção de comandos da controladora para voo autônomo
- Controle dos motores elétricos
- Realização do auto-nivelamento do quadricóptero
- Criação de missões de voo autônomos para o drone executá-las

## 2.3 Plataformas de Desenvolvimento

### 2.3.1 Mission Planner

Uma estação de controle em terra é imprescindível para que o veículo aéreo seja monitorado, sendo assim, o software Mission Planner e a placa ArduPilot em conjunto são responsáveis por inúmeras funções de navegação do quadricóptero, além de oferecer total interação entre o operador e a aeronave (CORRÁ e SILVA, 2014).

Fundado por Michael Osborne, esse sistema oferece suporte para aquisições dos dados em voo, verificação dos valores de output dos sensores em tempo real, download dos arquivos log para análise posterior, além de configurar a placa APM, escolhendo as estruturas de voo, dos comandos de rádio e as afinações dos valores controladores PID (Propositional Integral Derivate) (ANDRADE, 2015).

A figura 4 apresenta a interface do Software Mission Planner com uma rota planejada, com waypoints em amarelo.

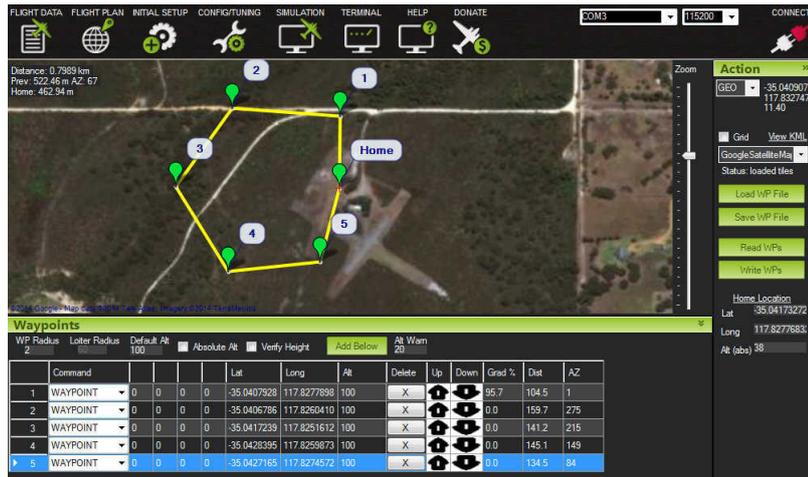


Figura 4 – Interface do Software Mission Planner

Os waypoints são as coordenadas geográficas definidas pelo operador que formam a trajetória do drone.

Conheça as principais funcionalidades do Mission Planner:

- Realizar o planejamento da missão
- Conectar o firmware (software) de piloto automático (APM)
- Monitorar o status do seu veículo durante a operação
- Gravar o log da telemetria

A figura 5 apresenta os dados de voo que são apresentados no Mission Planner, como Altitude em metros e Velocidade Vertical em metros por segundo.



Figura 5 – Dados de voo apresentados no Software Mission Planner

### 2.3.1.1 Programação em Arduino

Também é possível criar novas funcionalidades ou incorporar novos sensores no sistema a partir da programação em Arduino, nesse caso, a placa APM é programada a partir da IDE (Integrated Development Environment). Sendo assim, é necessário:

- Instalar a IDE do Arduino que se encontra disponível online e gratuitamente no site do próprio (presente nas referências).
- Baixar o firmware do Arducopter, disponível online e gratuitamente no site do próprio (presente nas referências).
- Clicar em ficheiro > abrir e procurar dentro da pasta ArduCopter o ficheiro “ArduCopter.pde”
- No menu Ferramentas escolhe-se a placa Arduino Mega 2560, que é a APM
- Efetuar as alterações desejadas no código, compilá-lo e, posteriormente, realizar o upload. Esse processo demora de um a dois minutos e a IDE avisa quando estiver completo.

A figura 6 apresenta o ambiente de desenvolvimento do Arduino, a IDE.

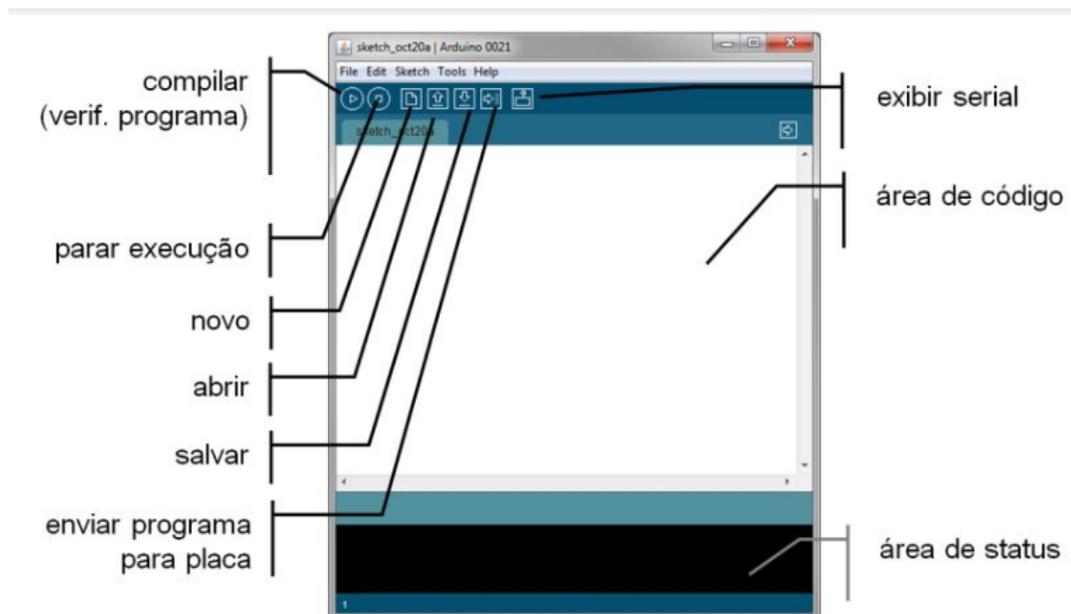


Figura 6 – IDE do Arduino

## 2.4 Sensores Essenciais para o Voo autonomo

### 2.4.1 Unidade Inercial

#### 2.4.1.1 Acelerômetro

O acelerômetro é um sensor capaz de medir forças de aceleração em um corpo em relação à gravidade, ou seja, pode medir vibrações, inclinações e colisões no veículo aéreo. Os acelerômetros utilizados em drones medem as forças em três eixos, e para aumentar sua eficiência é necessário que sua localização seja o mais próximo possível do seu centro de gravidade (VIEIRA, 2011).

#### 2.4.1.2 Giroscópio

Este dispositivo permite medir a velocidade de rotação de um objeto em torno do seu eixo de rotação, ou seja a sua velocidade angular. Assim, realizando a integral da velocidade em relação ao tempo, é possível saber a posição angular nos eixos, Yaw, Pitch e Roll.

Para o presente trabalho, é utilizado o chip MPU6000 que já vem embarcado na APM. Ele contém os dois sensores, acelerômetro e giroscópio, possuindo leitura pros 3 eixos do acelerômetro e 3 eixos para o giroscópio, atingindo 6 graus de liberdade. De acordo com CALDEIRA (2016), o chip lê as variações de ângulo sofridas pelo sensor, essas variações são sinais elétricos que são interpretados, convertidos e calculados pelo próprio MPU6000, que contém seu próprio processador, e ao Arduino é entregue os valores angulares.

O MPU-6000 possui internamente um recurso chamado DMP (Digital Motion Processor). O DMP possibilita que o algoritmo de detecção de movimento nos eixos do giroscópio e acelerômetro seja processado no próprio CI, permitindo que o ArduPilot utilize as saídas como em um sensor comum (SOUZA, 2015).

Esse sensor, possui pinos SCL (serial clock) e SDA (serial data) onde se dará a comunicação I2C entre ele e o microcontrolador do ArduPilot. Este protocolo de comunicação passa informações através de barramentos, isso permite que se possa capturar todos os valores de movimentação dos 6 eixos com a utilização de apenas duas entradas do microcontrolador.

A saída SDA do MPU-6050 é a responsável pelo fluxo de dados entre o mestre (ArduPilot) e o escravo (MPU6000), enquanto a saída SCL é responsável por receber a temporização advinda do microcontrolador, de modo que a comunicação pela SDA possa ter confiabilidade (REIS, 2014).

## 2.4.2 Controlador Eletrônico de Velocidade

O controlador de velocidade é responsável por fazer a associação dos sinais PWM (Pulse-Width Modulation) e a potência elétrica que deve ser fornecida para cada motor. Os sinais PWM são tensões, calculadas previamente pela APM, que modulam a largura do pulso que determina a intensidade de aceleração do motor de acordo com o comando do rádio-controle ou com o comando da trajetória de voo. Desse modo, o ESC converte uma sequência de dados digitais recebida em sinais de corrente alternada que será fornecida aos motores, possibilitando assim a movimentação desejada do quadricóptero (SÁ, 2012). A Figura 7 exemplifica como um sinal de tensão PWM é quantificado para a associação com a potência do motor.

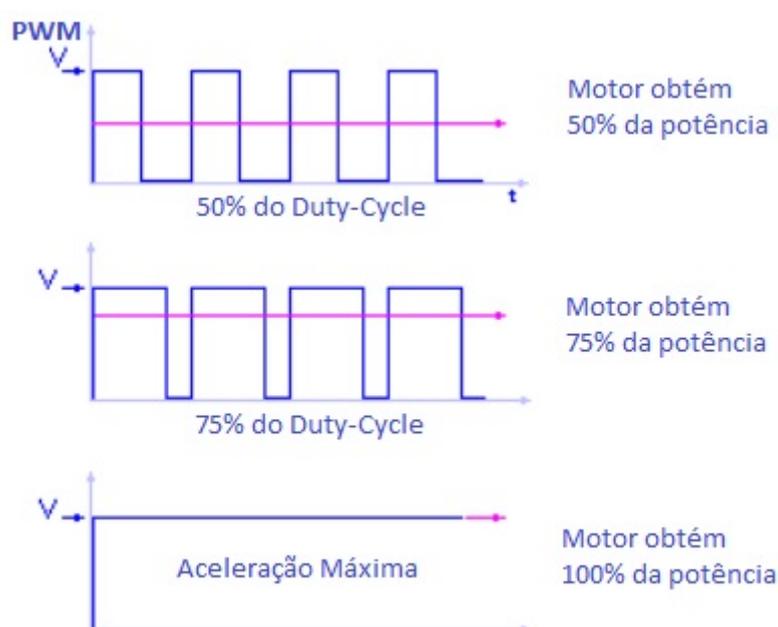


Figura 7 – Relação dos sinais PWM e a potência do motor

Um ESC é composto por um conjunto de transistores, que são ligados ou desligados determinando assim a quantidade de corrente a ser enviada pro motor, e um microcontrolador, que interpreta as entradas e elabora o controle de acordo com um software existente. O sistema de envio de dados para o motor deve ser muito eficiente quanto à precisão e ao tempo de atuação, pois se os motores não atingirem a velocidade adequada para o controle, a estabilidade do sistema será prejudicada.

## 2.4.3 GPS (Global Positioning System)

### 2.4.3.1 Funcionamento

O GPS (Global Positioning System) é um sistema de satélites que tem como função enviar informações sobre a posição de um determinado objeto em qualquer lugar do globo.

O aparelho receptor (o dispositivo GPS) capta as informações de um grupo de quatro satélites e, através da troca de alguns dados e algoritmos, consegue determinar para o utilizador a sua exata localização no mapa.

Esse sistema utiliza um processo que mede distâncias chamado de trilateração, onde cada satélite possui sua posição conhecida no espaço, como mostra simplificada em duas dimensões a figura 8.



Figura 8 – Satélites no espaço

Os satélites que estão próximos ao receptor ficam enviando sinais de rádio continuamente, esses sinais são ondas portadoras que contém o instante de tempo em que aquele pulso foi emitido e a posição real do satélite. Aproveitando-se da distância medida, o satélite a utiliza como raio para criar um círculo e a posição do GPS pode estar localizada em qualquer lugar da circunferência, como mostra a figura 9.

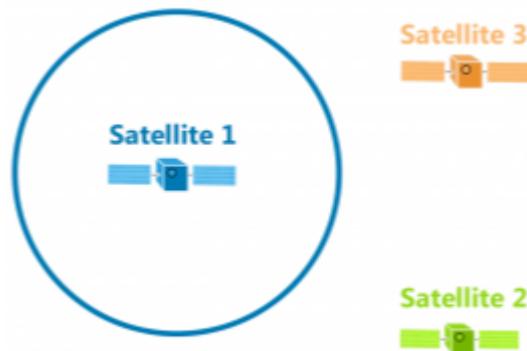


Figura 9 – Círculo formado pelo satélite

Para determinar a distância entre o satélite e o receptor, um cálculo é realizado: a comparação de tempo em que o sinal foi enviado, com o tempo exato em que ele foi recebido, sempre considerando que a radiofrequência viaja na velocidade da luz.

Esse processo é realizado com mais 3 satélites e, posteriormente, cruzando as informações de posição de todos, obtêm-se a posição do receptor no lugar em que as circunferências se encontram, como mostra a figura 10. Além da latitude e longitude, o sistema

de trilateração também permite saber a altura do receptor em relação ao nível do mar (MACHADO, 2012).

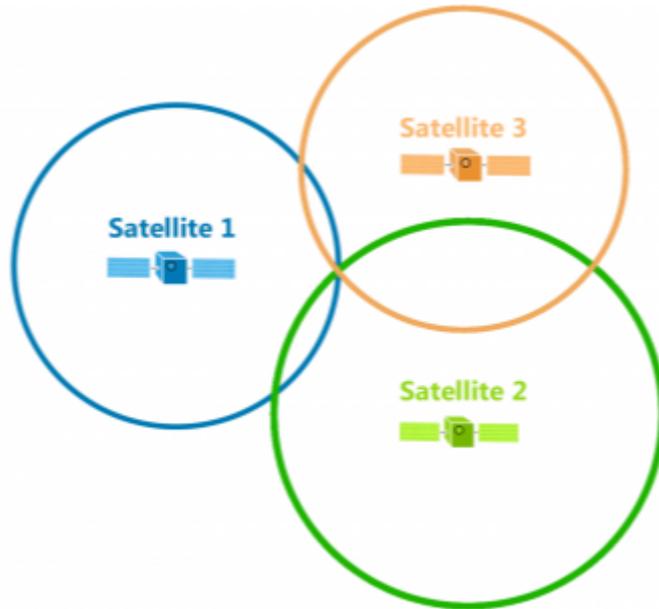


Figura 10 – Encontro das circunferências dos satélites que irá definir a posição do receptor GPS

O modelo bidimensional foi utilizado para facilitar o entendimento, porém o sistema é realizado em três dimensões, onde os satélites que transmitem sinais os sinais de GPS utilizam-se de uma esfera, não de uma circunferência (Gisgeography, 2018), como apresentado na figura abaixo.

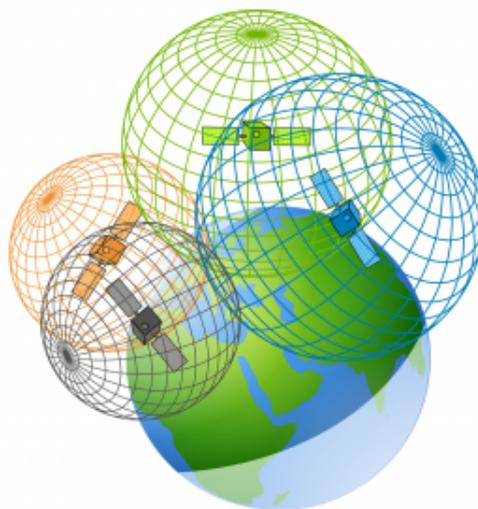


Figura 11 – Modelo real tridimensional do GPS

### 2.4.3.2 Aplicação em um veículo aéreo

O GPS tem a função de determinar qual a localização do VANT e é imprescindível para o voo autônomo, pois é a partir dos seus dados de posição que o drone saberá sua real posição e pode conferir se está na rota pré-programada ou se precisa corrigi-la.

O dispositivo utilizado neste projeto é o módulo Ublox modelo NEO 6M, pois é totalmente compatível com a APM e além de ser um módulo com GPS, possui também uma bússola. A calibração do GPS é automática após a inicialização da placa controladora, portanto não há nenhuma necessidade de calibração posterior. A Figura 12 apresenta o módulo usado, a APM e a ligação entre eles.

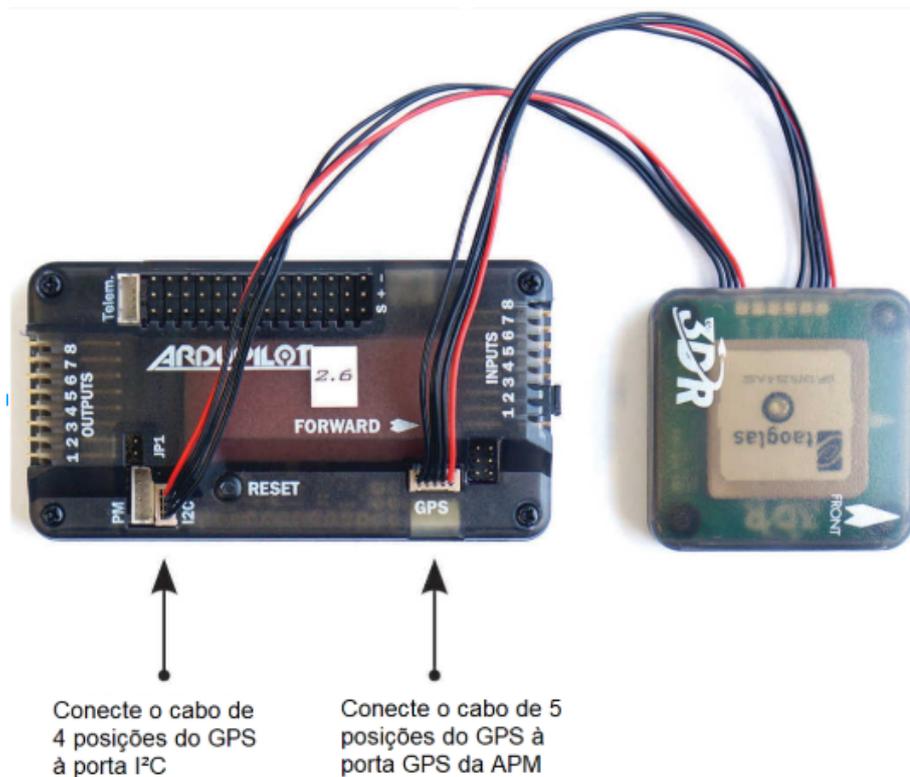


Figura 12 – Ligação entre a APM e o módulo GPS Ublox

### 2.4.4 Bússola

A bússola é a responsável pela orientação dos pólos para o veículo aéreo, ou seja, é ela que é a principal fonte de informações de rumo do seu drone. Sem um rumo preciso, o veículo não se moverá na direção correta nos modos de piloto automático. Sendo assim, a configuração precisa da bússola é fundamental para que ocorra o voo desejado e com segurança (ARDUPILOT, 2016).



Figura 13 – Módulo Ublox na haste

Para ser colocada no drone, a bússola necessita de uma base com uma haste longa para deixá-la afastada dos outros componentes, pois eles criam um campo magnético que causa interferência na leitura da bússola, qualquer interferência magnética nesse dispositivo pode ter um sério impacto na navegação.

A figura acima apresenta a haste que fixará o módulo que contém o GPS e a bússola na estrutura do veículo.

Analisando-se o datasheet do dispositivo, é possível encontrar a precisão dele para medidas de velocidade e posição no eixo horizontal, que são, respectivamente, 0,05 m/s e 2,5 m.

#### 2.4.5 Barômetro

O barômetro é um instrumento utilizado para aferir pressões atmosféricas, no VANT é ele quem mensura a variação de pressão, provocada pela altitude, e assim indica para controladora a altura do drone, funcionando assim como altímetro.

Conhecer em qual altura o drone está é essencial para o voo autônomo e para rota que ele seguirá, da mesma forma que nos modos de voo com piloto automático que demandam que a altitude seja “travada” é ele quem fornece essa informação continuamente para correção dos erros de altitude. A figura 14 apresenta a localização deste sensor na placa e como ele é encontrado no mercado.

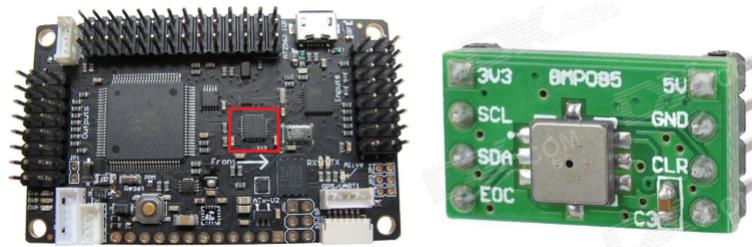


Figura 14 – À esquerda: barômetro embarcado na placa APM, à direita: barômetro digital

Este sensor é extremamente sensível à luz e à vibrações, sendo assim, é necessário que mesmo embarcado na APM ele seja protegido destas variáveis.

#### 2.4.6 Motores e Hélices

Para propulsão dos veículos aéreos, geralmente, usam-se motores brushed (com escovas) ou motores brushless (sem escovas), porém os motores brushed são de baixa performance, e os brushless se fazem mais populares no aeromodelismo por causa de sua alta rotação, força e conseqüentemente poder levantar uma quantidade maior de peso, atingindo maiores alturas (MOLINA, 2015).

Os motores brushless são motores de corrente contínua compostos de um rotor e um estator. No rotor estão presentes os ímãs permanentes, que não necessitam de alimentação alguma pois são ativados por campo magnético que é gerado a partir das bobinas presentes no estator. O movimento então é produzido quando o estator, que é fixo, é alimentado e gera campo magnético nas bobinas, responsável por fazer o rotor girar (TRINDADE, 2009). A Figura 15 apresenta o tipo de motor brushless utilizado neste projeto, que é um motor que possui o estator fixo no meio da estrutura, fazendo o rotor, que é externo, girar.

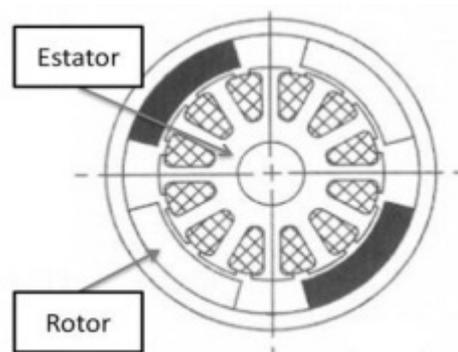


Figura 15 – Motor Brushless com rotor externo

Como o drone depende unicamente da força dos motores (e especialmente dos efeitos de rotação dos mesmos) para manutenção do seu voo, este deve ser o primeiro

ponto de análise. Contudo, não se pode estimar a propulsão necessária sem saber, por exemplo, o peso total da aeronave. Portanto, a escolha dos motores segue um padrão de previsão.

Para se obter um parâmetro de segurança para o bom funcionamento do motor utiliza-se a seguinte fórmula:

$$E = \frac{2P}{n} \quad (2.1)$$

Onde:

E = Empuxo necessário para cada motor

P = peso total sem motores

n = número de motores

Tabela 2 – Peso Estimado do Drone

Peça	Quantidade	Peso
Bateria	1	358g
Frame	1	282g
Motor	4	192g
Hélices	4	45,6g
GPS	1	32g
APM	1	30g
Receptor de Rádio	1	6,4g
Telemetria	1	40g
ESCs	4	100g
PowerModule	1	20g
Outros	-	100g
Total		1,206kg

Então, sabendo o peso do quadricóptero, a partir do empuxo de cada motor é possível saber se estes serão capazes de realizar um voo eficiente. Portanto, para especificações deste projeto, que tem peso total estimado em 1.217 quilos apresentado na tabela 2, localizada na próxima página, foi escolhido o motor 2212 da DJI que possui capacidade de realizar 920 rpm/V, tem diâmetro de 22 e altura de 12 milímetros, pois é compatível com a bateria utilizada e nos testes de empuxo realizados posteriormente, apresentou bom desempenho.

As hélices precisam ser escolhidas de acordo com seu sentido de rotação, sendo assim, foram escolhidos dois pares, um usado para girar em sentido horário e o outro usado no sentido anti-horário. Assim, dois rotores giram em um sentido e os outros dois em sentido reverso. A correta disposição dos motores elimina o efeito dos torques gerado por eles. Logo, hélices adjacentes devem girar em sentidos opostos e assim, um motor

elimina o efeito do torque do outro fazendo com que o drone fique em equilíbrio em uma determinada posição, quando os motores tiverem mesma velocidade angular (SÁ, 2012).

A Figura 16 apresenta as hélices utilizadas, que foram escolhidas baseadas nos motores escolhidos anteriormente, para o nosso projeto, decidiu-se utilizar as hélices DJI 9443, estas são de duas pás, de ótima qualidade e com grande compatibilidade com o motor.



Figura 16 – Hélices DJI 9443 E Motores 2212 DJI

#### 2.4.7 Rádio controle e receptor de comandos

Para que ocorra o controle remoto do drone antes da ativação da missão, precisa-se de um transmissor de comandos (rádio controle), tanto quanto de um receptor desses comandos. Abaixo, segue tabela comparativa de preços dos rádio-controles mais populares:

Tabela 3 – Dados comparativos entre os rádio-controles mais comumente utilizados

Modelo	Processador	Clock do Processador	Valor
FlySky TH9X	8 bits	16 MHz	256,21
RadioLink AT-9	32 bits	72 MHz	288,24
FlySky FS-i6S	32 bits	48 MHz	192,15
FlySky FS-i6	32 bits	48 MHz	138,00

Para atender às especificações da controladora, o modelo FlySky FS-i6 foi escolhido, Figura 17, por possuir processador e frequência de clocks satisfatórias e aplicáveis ao projeto, tanto quanto pela relação custo-benefício, este rádio tem alcance de até 1,6 km horizontal e 300 metros na vertical.



Figura 17 – Rádio Controle e receptor FS-i6

### 2.4.8 Estrutura

A peça que determina as características principais da estrutura do drone é chamada de frame. A partir de suas informações conhecemos a distância entre os braços, e as características do material nos dá informações sobre resistência, peso e vibração da estrutura. Optou-se para o projeto pela utilização de um frame comercial para compor a estrutura do drone. O frame em questão, modelo F450, é o mesmo utilizado em modelos de drones da marca Phantom, portanto tem a garantia de ser de boa qualidade e funcional, o que ajuda bastante na montagem do quadricóptero e na qualidade do voo.

Este modelo, mostrado na Figura 18, traz algumas vantagens, a placa central tem 2 funções, ela dá rigidez ao drone e também conduz a eletricidade necessária para todos os componentes eletrônicos, assim, minimizando o número de fios. O material que constitui os braços é de plástico PA66 30GF, resistente e minimiza as vibrações. A distância entre eixos é de 450mm e o peso é de 282g.



Figura 18 – Frame F450

### 2.4.9 Alimentação

A alimentação dos quadricópteros atuais é realizada por baterias do tipo LiPo (polímero de lítio) pois possuem uma maior densidade de energia comparada com as

baterias de níquel, oferecendo uma maior duração de carga e um peso menor, o que é essencial para um voo eficaz. Ela pode também ser carregada sempre que necessário. A maioria das baterias deste tipo, usa uma carga rápida para carregar o seu equipamento a 80% da capacidade da bateria e então muda para carga lenta. A bateria irá alimentar tanto os ESC's quanto o Arduino. A partir da escolha do motor e da análise de desempenho, optou-se pela bateria Power Lipo 5200 mAh 3s, que fornecerá uma tensão de 11,1 V para o sistema, Figura 19.



Figura 19 – Bateria de lítio da marca Tattu utilizada no projeto

### 3 Teoria de Operação do Quadricóptero

Um quadricóptero é formado basicamente de quatro propulsores organizados de maneira cruciforme. Cada uma das hélices é responsável por uma parcela do empuxo total exigido para decolagem e sustentação do veículo (PFEIFER, 2013). De forma a equalizar os momentos gerados por cada propulsor, um par gira no sentido horário e o outro no sentido contrário, desta maneira, se todos os propulsores estiverem com velocidades angulares equivalentes, o somatório dos momentos angulares será zero. Aproveitando esta situação, é possível também desequilibrar os torque para criar uma movimentação, ou seja, cada motor além de criar forças verticais também introduz também forças horizontais.

RESENDE (2014) afirma que o quadricóptero tem um sistema que apresenta 6 graus de liberdade, sendo capaz de realizar translação vertical no eixo Z, horizontal lateral no eixo Y, horizontal longitudinal do eixo X e rotação de rolagem (ângulos em torno do eixo X), arfagem (ângulos em todos do eixo Y) e guinada (ângulos em torno do eixo Z), como é apresentado na Figura 20.

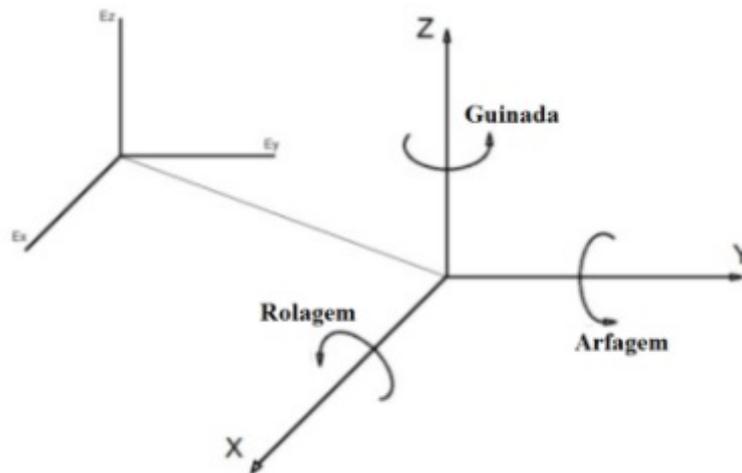


Figura 20 – Graus de liberdade do quadricóptero

Neste sentido, o movimento do quadricóptero em qualquer um dos graus de liberdade demanda aceleração e desaceleração em algum dos rotores, para facilitar a compreensão dos esquemas dos movimentos, as figuras a seguir, apresentarão a rotação dos motores, o esquema de movimentos e os eixos, onde  $\Omega_c$  é a velocidade angular de cada um e  $\Omega_4$  é o motor frente-esquerda,  $\Omega_3$  é traseira-esquerda,  $\Omega_1$  é o frente-direita e  $\Omega_2$  é o traseira-direita.

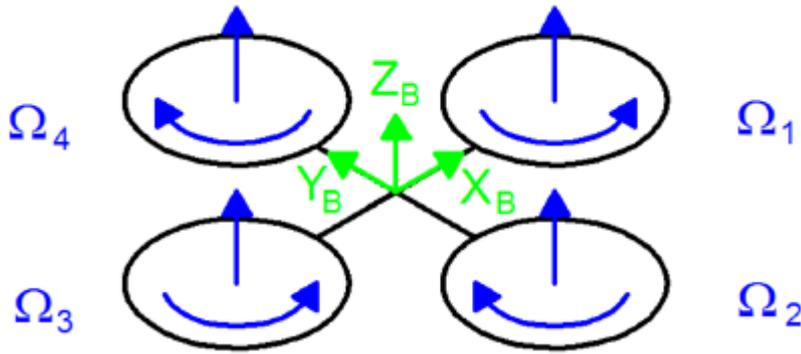


Figura 21 – Princípio de Funcionamento

Quando a velocidade de rotação dos motores ( $\Omega_c$ ) é incrementada, ampliam-se assim as forças de elevação do veículo, nesse sentido, o movimento mais simples de ser realizado é o vertical. Se as acelerações ( $\Delta a$ ) aplicadas nos quatro motores forem de mesma amplitude, o veículo irá deslocar-se verticalmente de baixo para cima (sobe) se a aceleração ( $\Delta a$ ) for positiva (Figura 22), ou irá deslocar-se verticalmente de cima para baixo (desce) se a aceleração for negativa.

Isso significa que as velocidades nos rotores irão aumentar ou diminuir de acordo com  $\Omega_c + \Delta a$ .

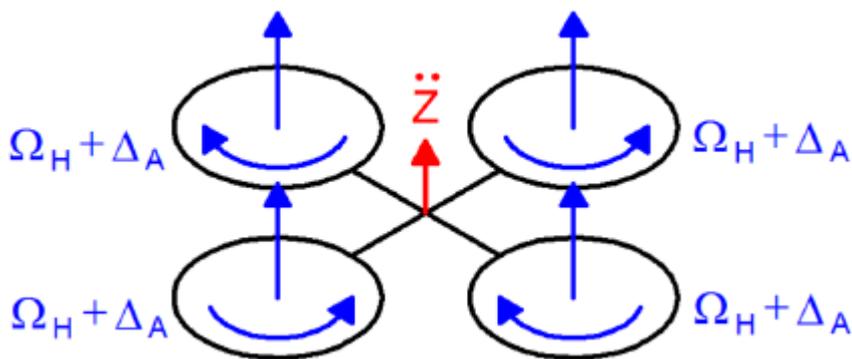


Figura 22 – Movimentos verticais

Caso o drone queira realizar o movimento de guinada, também conhecido como Yaw, ele necessita fazer a plataforma girar em torno do eixo vertical no sentido horário, sendo necessário aumentar a velocidade de rotação do par de motor 2 e 4 com uma aceleração  $\Delta a$  e diminuir a velocidade de rotação dos motores 1 e 3 com uma desaceleração  $\Delta b$ , para realizar o movimento anti-horário, o que é apresentado na Figura 23, caso o movimento desejado seja virar para direita, os motores terão as velocidades trocadas.

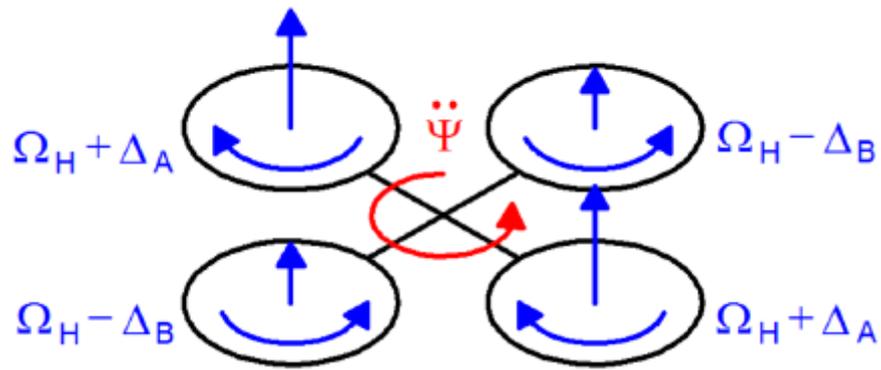


Figura 23 – Movimento Yaw

Nesse modelo, aos rotores situados em torno do eixo YB, soma-se uma variável  $\Delta a$ ,  $(\Omega_c + \Delta a)$  e subtrai-se uma variável  $\Delta b$  em torno do eixo XB  $(\Omega_c - \Delta b)$ .

Para atuar a arfagem ou inclinação, também conhecido como Pitch, o par de motores 1 e 4 mantém a sua velocidade de rotação constante, enquanto os motores 2 e 3 aumentam a sua velocidade de rotação em  $\Delta a$  para o veículo aéreo movimentar-se para trás, Figura 24, assim como, se o movimento desejado for seguir em frente, as velocidades dos motores são invertidas.

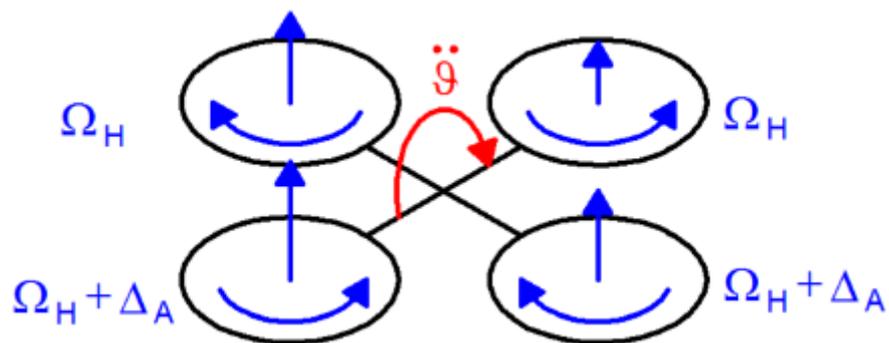


Figura 24 – Movimento Pitch

Por fim, a atuação da rolagem, também conhecido como Roll, segue o mesmo procedimento de Pitch, mas neste caso o par de motores que irá manter-se fixo é o 1 e o 2, enquanto o par 4 e 3 altera a sua velocidade de rotação e o drone curva-se para direita, esta condição dá origem à situação da Figura 25 e, caso aumente a rotação dos motores 1 e 2 e mantenha a velocidade dos motores 3 e 4, o drone curva-se para esquerda.

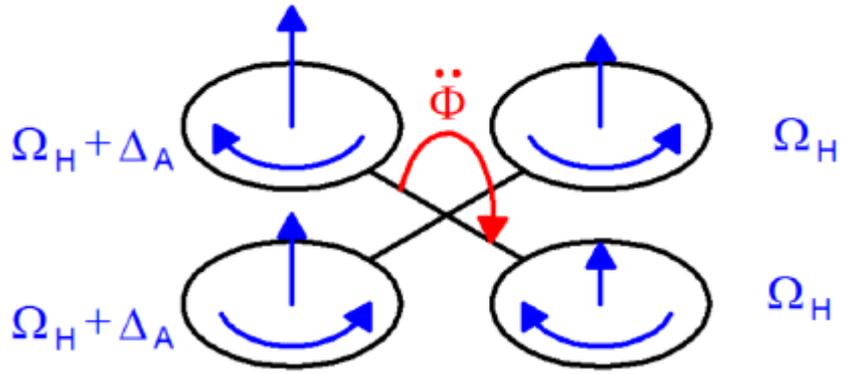


Figura 25 – Movimento Roll

Sendo assim, é possível perceber que o veículo aéreo pode ser movimentado em qualquer direção no espaço utilizando-se da associação de movimentos (AFONSO, 2014).

## 4 Modos de Voo

Para que o veículo aéreo seja capaz de voar autonomamente é necessário que este seja capaz de realizar o controle de sua atitude, altitude e de sua posição. O controle de atitude é realizado em qualquer modo de voo que seja e é baseado nos controles de parâmetros PID (Proportional, Integral and Derivate).

No entanto, existem modos de voo específicos para o controle de altitude e de posição, sendo assim, o Modo Automático é uma incorporação destes dois e qualquer um dos três modos, controle de altitude, posição ou automático, são acionados a partir do controle remoto ou da estação de controle (Mission Planner).

Primeiramente, será descrito o controle de atitude do drone e, posteriormente, os modos de voo de controle de altitude, posição e autônomo.

- Controle de Atitude

Para que um quadricóptero mantenha um voo condizente com o que é esperado, é necessário que este seja capaz de realizar ações para controlar suas atitudes ao mesmo passo que realiza seu autonivelamento.

Sendo assim, é imprescindível que a aeronave realize o controle de suas variáveis, apresentadas no capítulo anterior. O nome deste modo de voo é **Stabilize**, onde as atitudes do veículo podem ser requeridas ainda por um controle remoto (transmissor) ou estão incorporadas em outro modo de voo, por exemplo, podem ser atitudes requisitadas por uma missão.

O sistema de controle de um drone possui um algoritmo computacional que recebe um comando advindo da rota pré-definida, ou no caso de um voo controlado, advindo do transmissor. Este define o movimento que ocorrerá, lê esse sinal e atua na rotação dos motores com a precisão necessária para que ocorra o movimento esperado. Para atingir esse requisito, o controlador deve estar calibrado para a execução dos passos em tempo certo e ter o erro minimizado, senão o voo do veículo será prejudicado (AVELINO, 2016).

Como dito anteriormente, o controle de atitude é baseado nos controles de parâmetros PID (Proportional, Integral and Derivate).

- Controlador PID Para compensar as instabilidades do voo, uma das técnicas mais utilizadas é a de controle PID. Para Bresciani (2008) a popularidade deste tipo de algoritmo para sistemas de controle é proveniente de três características:

- \* Estrutura simples

- \* Boa performance em vários processos
- \* Possibilidade de afinação dos parâmetros sem conhecimento prévio do modelo matemático do sistema

Com as variáveis necessárias já adquiridas pela unidade inercial MPU6000, que são as medidas angulares de onde o drone se encontra, o papel do controlador PID é modelar a ação do controle, operando sobre o sinal de erro (SOUSA, 2001). O erro é a diferença entre o ângulo que se deseja e o ângulo que foi medido, dessa forma, o controle de ângulo em malha fechada para cada eixo do quadricóptero baseia-se no sinal de erro. O controle do ângulo está esquematizado na figura 26.

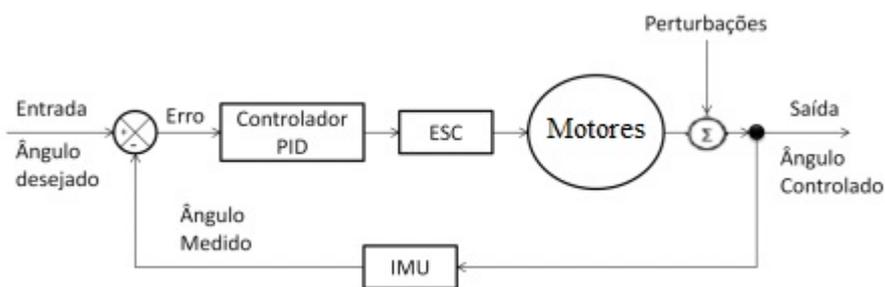


Figura 26 – Diagrama de Blocos em malha fechada

De acordo com Cavalcante (2012), o sinal contínuo de erro na entrada do controlador é amostrado e convertido em sinais digitais, enquanto a saída do controlador é convertida num sinal analógico contínuo alimentando o processo, onde as entradas do controlador são as posições calculadas pelo código responsável e os valores medidos pelos sensores e a saída do controlador contém os valores ajustados dos ângulos.

$$u(t) = K_p * e(t) + \frac{K_p}{T_i} \int e(t) dt + K_p T_d * \frac{d e(t)}{dt}$$

Figura 27 – Fórmula que descreve o controlador PID

Como é possível perceber na Figura 27, a saída do controlador PID é uma combinação de três ações e cada uma reage de forma distinta ao erro presente nos sistemas, de forma que o controle proporcional ajusta a variável de controle de forma proporcional ao erro multiplicado por um ganho  $K_P$ , o controle integral ajusta a variável de controle baseando-se no tempo em que o erro acontece multiplicado por um ganho  $K_I$  e o controle derivativo ajusta a variável de controle tendo como base a taxa de variação do erro multiplicado por um ganho  $K_D$ .

- Controle de Altitude

O nome do modo de voo que realiza o controle de altitude é **AltHold**, neste modo o veículo aéreo mantém uma altitude enquanto permite que o roll, pitch e yaw sejam controlados normalmente de forma remota.

Como dito anteriormente, os modos de voo são acionados a partir do controle remoto ou da estação de controle, então, o AltHold, quando selecionado, controla o acelerador do drone automaticamente para manter a altitude atual. Roll, pitch e yaw podem ainda ser controlados da mesma maneira que no modo Stabilize, o que significa que o piloto pode controlar ainda os ângulos de inclinação, de rolamento e o rumo do drone. Para este modo de voo é essencial que o barômetro esteja funcionando perfeitamente, pois é ele que determina a altitude em que o drone está a partir da pressão do ar.

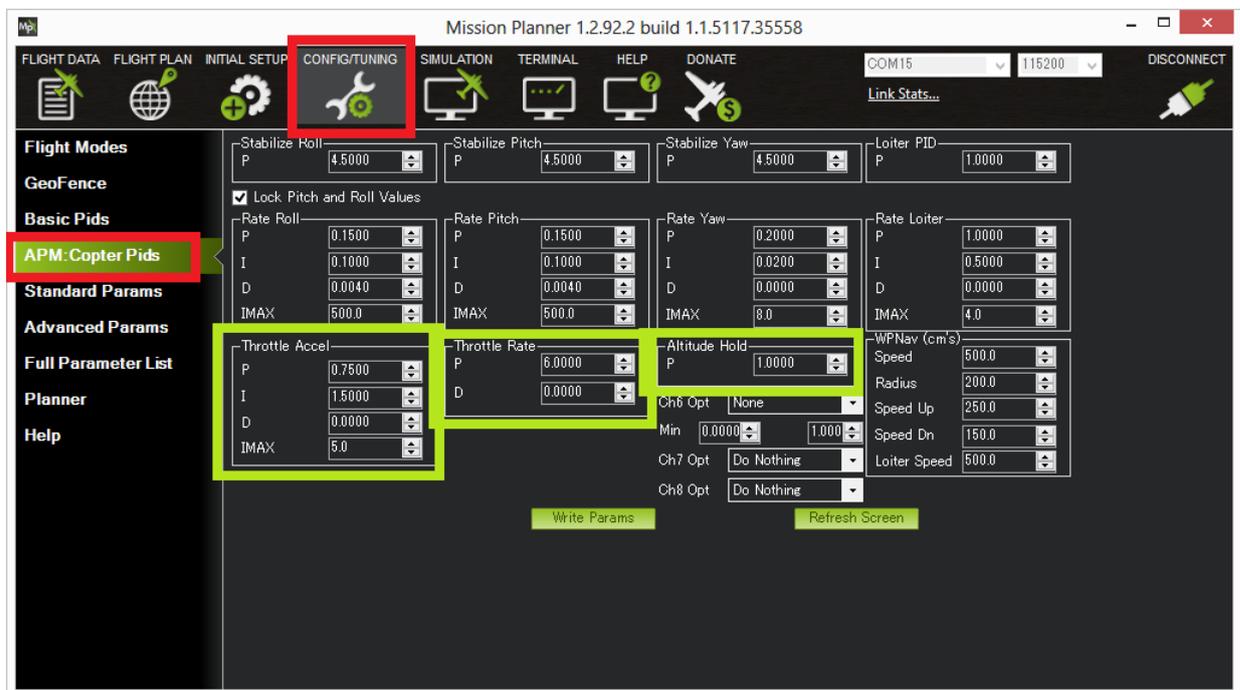


Figura 28 – Parâmetros do modo AltHold

Como é possível perceber na Figura 28, existe um parâmetro P para o Altitude Hold, ele é usado para converter o erro de altitude, que é a diferença entre a altura desejada e a real, para uma taxa de subida ou descida desejada. Por exemplo, quanto mais alto for o valor de P, mais agressiva a taxa com que o drone tenta manter sua altitude é. Uma taxa muito alta pode acarretar em resposta muito brusca do acelerador e, assim, desestabilizar o voo.

A taxa de aceleração converte a taxa de subida ou descida desejada em uma aceleração desejada para cima ou para baixo, normalmente não requer nenhum ajuste.

Os ganhos do PID do Acelerador (Throttle Accel) convertem o erro de aceleração (ou seja, a diferença entre a aceleração desejada e a aceleração real) em uma saída do motor. É recomendável, assim como apresentado na Figura 28, que a proporção 1: 2 de P para I seja mantida caso haja alguma modificação de valores.

A Figura 29 apresenta todo o sistema de ajuste das variáveis que foi apresentado acima, onde a taxa de mudança de altitude (o parâmetro P do AltHold) é intrinsecamente ligada ao controle PD do Throttle Rate, que é aceleração de subida ou descida do drone, e do controle PID do Thottle Accel, que realiza o controle de aceleração angular do sistema todo em si, como no modo Stabilize, a partir de dados dos ângulos da unidade inercial e o envia para os motores.

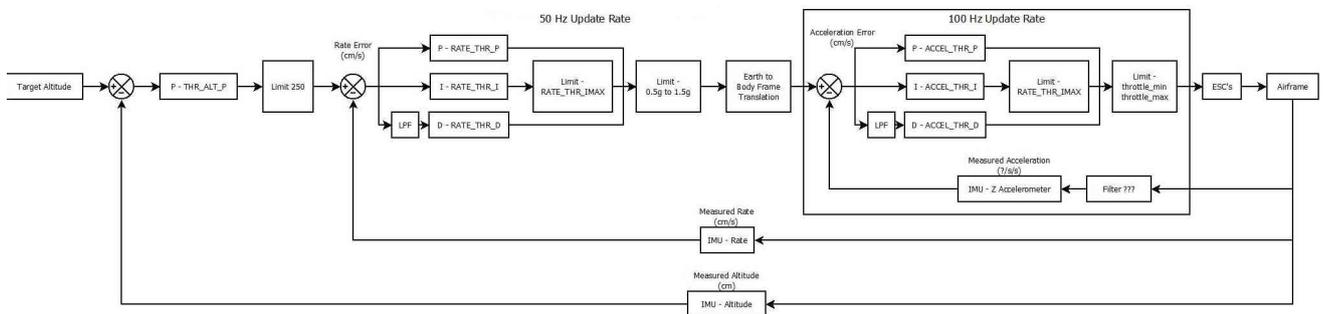


Figura 29 – Controle PID do modo AltHold

- Controle de Posição

O modo de voo que realiza o controle de posição é chamado **Loiter**, onde o veículo aéreo tenta manter automaticamente sua localização atual, ou seja, tanto sua altitude quanto seu rumo ficam parado em uma posição.

Para o loiter ser realizado com sucesso é necessário que o barômetro esteja em ótimas condições para medição da altitude, assim como, não haja interferências magnéticas consideráveis na bússola para que o rumo do drone seja mantido e o GPS tenha uma boa trava, para manter o drone na exata localização geográfica desejada.

O modo Loiter incorpora o controlador de altitude do modo AltHold, sendo assim, existem controladores PID para manter a altitude do drone, explicados na seção acima, e ainda controladores para manter o rumo.

Primeiramente, é necessário um controle para a posição, então, o parâmetro P, indicado na figura 30 como “Loiter PID P” no canto superior direito, é responsável pela conversão de erro da posição horizontal, ou seja, a diferença entre a posição desejada e a posição real, em uma velocidade desejada em direção à posição desejada.

Posteriormente, um outro controlador PID é responsável por converter a velocidade desejada em direção ao alvo para uma aceleração desejada, esta irá tornar-se um

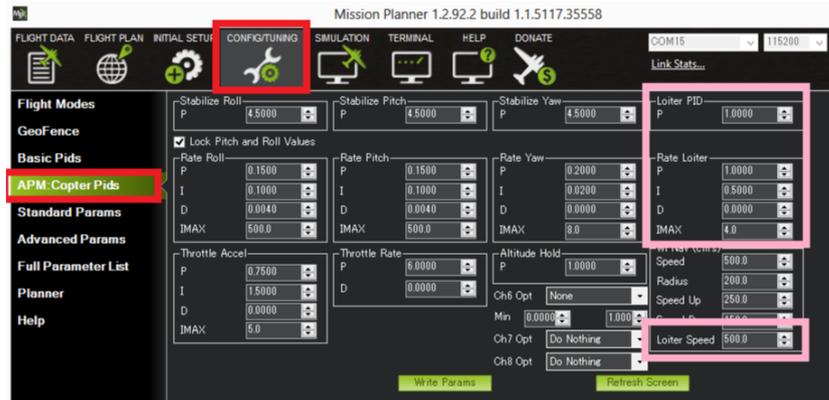


Figura 30 – Parâmetros do Loiter

ângulo de inclinação que será passado para o mesmo controlador angular usado no Stabilize e no final do modo AltHold.

- Modo Autônomo

O modo autônomo, ou automático, é conhecido pelo nome **Auto**. Neste modo, o veículo aéreo segue um script de missão pré-programado e armazenado na controladora de voo. Este script é composto por vários comandos: os comandos de navegação, por exemplo, os waypoints, que são as localizações geográficas desejadas na rota, e comandos do tipo “DO”, que não afetam a localização do veículo mas realizam funções auxiliares. A figura a seguir ilustra didaticamente como uma missão será realizada neste modo.

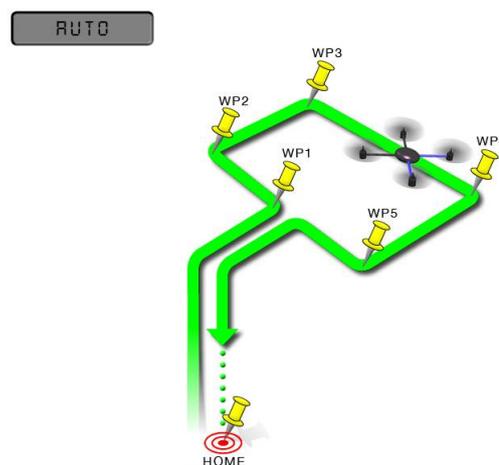


Figura 31 – Waypoints

Como apresentado anteriormente, o modo Auto é a incorporação dos modos de voo AltHold e Loiter, ou seja, ele só funcionará se os controles de altitude e posição estiverem funcionando.

Diferente dos modos de voo Loiter e AltHold, o modo Auto não admite controles externos, senão os da controladora de voo a partir da missão pré-programada, o que um piloto pode fazer é, no máximo, trocar a chave de posição no controle para o modo Stabilize ou requisitar na base de controle que o modo de voo seja mudado para Stabilize para ele readquirir o controle do drone.

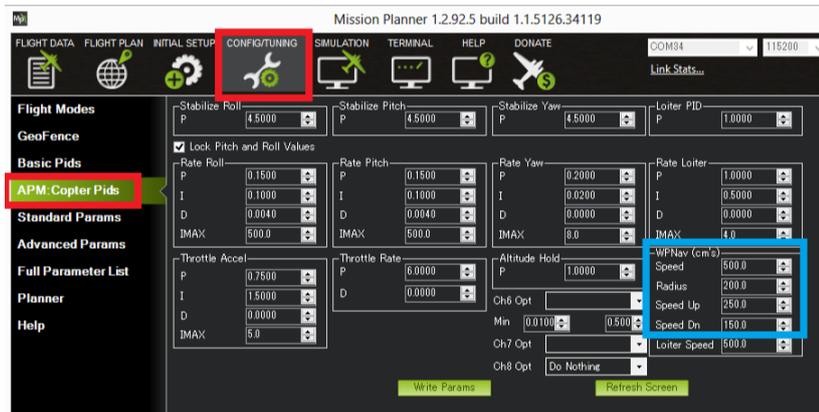


Figura 32 – Parâmetros do modo Auto

Como é possível perceber na figura 32, o modo auto não necessita de controladores PID sem ser os já advindos dos modos loiter e althold, o que pode ser modificado neste modo:

- A velocidade horizontal (Speed) com que o drone se encaminhará para o próximo waypoint
- As velocidades verticais (Speed\_Up e Speed\_Dn) com que o drone se moverá para cima ou para baixo
- O quão próximo do waypoint o drone chega para que este seja considerado “completo” e o drone poder ir para o próximo, dado pela função Radius. Isso só acontece quando um atraso é incluído no waypoint, senão, o drone segue normalmente até o waypoint.

## 4.1 Análise de desempenho a partir de logs

Durante todo o voo, o veículo aéreo recolhe dados importantes sobre a navegação, o que é essencial para análise de desempenho deste para verificar se o drone está cumprindo com o desejado.

Então, após a realização de um voo, um registro de log de dados pode ser baixado e aberto com planejador de missões (Mission Planner). Desta maneira, o software é capaz de representar graficamente as variáveis de voo desejadas.

No caso do controle de altitude (AltHold), são representados no gráfico a altitude do barômetro, a altitude desejada e a estimativa de altitude baseada na navegação inercial, como mostrado na figura 33.

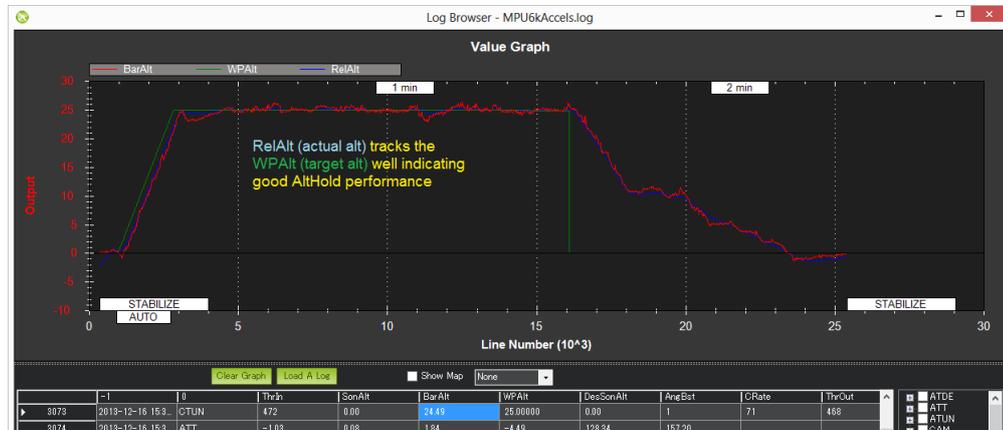


Figura 33 – Gráfico de desempenho do AltHold

No caso do controle de posição, são representadas graficamente as velocidades no eixo X, assim como as variáveis do modo AltHold, como é mostrado na figura 34.

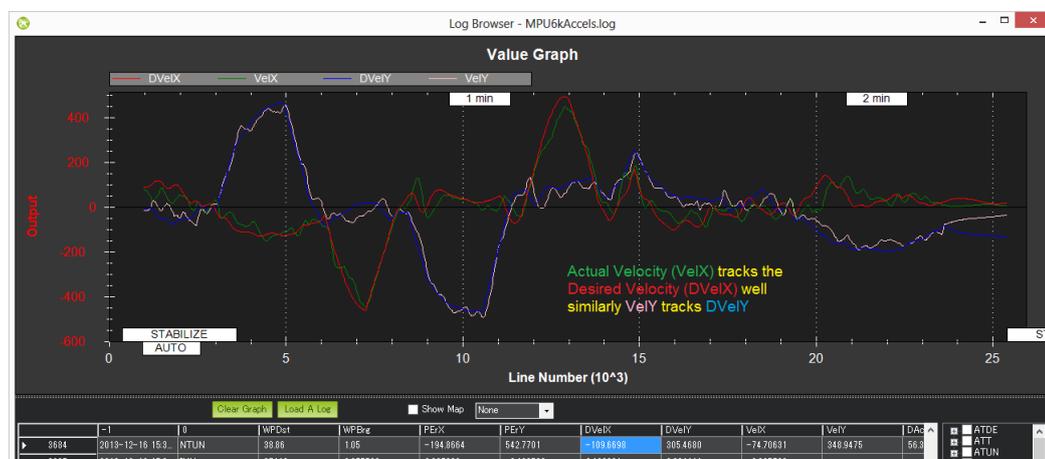


Figura 34 – Gráfico de desempenho do Loiter

A análise gráfica é bastante intuitiva, pois nela é possível perceber visualmente se as variáveis reais estão muito divergentes das esperadas.

## 4.2 Aquisição da Velocidade e Posição

Para uma análise de desempenho do veículo aéreo, é necessário que variáveis como altitude, velocidade e posição sejam adquiridas e analisadas. A altitude do drone é adquirida de forma direta a partir do barômetro, porém, a aquisição da velocidade e posição do drone é dependente dos sensores de GPS e acelerômetro.

O acelerômetro funciona em altas frequências, tem grande precisão, sendo sensível a pequenas variações, porém, seu sinal é sujeito ao efeito de deriva, o que ocasiona grande acumulação de erros durante o tempo. Já o GPS funciona em baixa frequência, tem pouca precisão, apresentada na Seção 2.4.4, mas em contrapartida é uma medida absoluta, não acumulando erros durante o tempo.

Como é mostrado na Figura 35, os sensores têm características diferentes e se completam, sendo necessário fazer a fusão de suas medidas para um bom funcionamento do controle das variáveis de velocidade e posição.

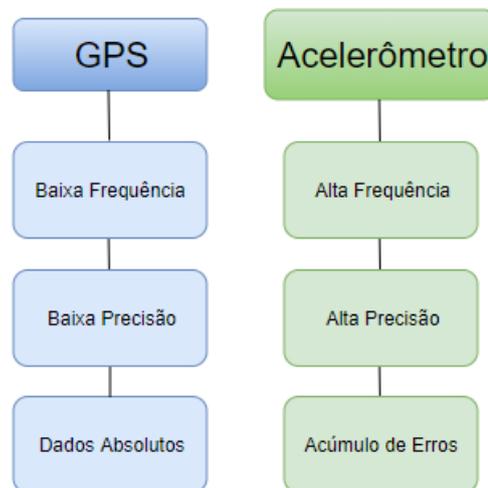


Figura 35 – Características complementares dos sensores GPS e acelerômetro.

A fusão destes sensores é feita utilizando o filtro de Kalman, que utiliza medidas de grandezas realizadas ao longo do tempo contaminadas com ruído e outras incertezas e gera resultados que tendem a se aproximar dos valores reais das grandezas medidas.

O filtro de Kalman pode ser dividido em duas etapas: predição e atualização. A predição é chamada assim pois utiliza as medidas do acelerômetro e a estimativa do passo anterior para fazer uma previsão do estado atual, sem levar em conta a informação vinda da observação do estado atual (medida realizada pelo GPS). A etapa de atualização utiliza a estimativa realizada na predição em conjunto com as medidas do GPS para refinar a estimativa do estado.

Para utilizar o filtro de Kalman é necessário modelar o processo de acordo com a sua estrutura. Isto é feito a partir da especificação das matrizes  $F_k$ ,  $B_k$ ,  $Q_k$  e  $R_k$ , para cada passo de tempo  $k$ .

- $F_k$ : Modelo de transição de estados, aplicado no estado anterior  $x_{k-1}$ .
- $B_k$ : Modelo das entradas de controle, aplicado no vetor de entradas de controle  $u_k$ .

- $Q_k$ : Covariância do ruído do processo.
- $R_k$ : Covariância do ruído da observação.
- $H_k$ : Modelo de observação.

O modelo para o filtro de Kalman assume que o estado atual no tempo  $k$ ,  $x_k$ , é obtido através do estado no tempo  $k-1$ ,  $x_{k-1}$ , de acordo com a seguinte equação:

$$x_k = F_k x_{k-1} + B_k u_k + w_k \quad (4.1)$$

Para esta aplicação  $F$ ,  $H$ ,  $R$  e  $Q$  são constantes e os índices temporais são ignorados. A matriz  $Q$  é obtida colocando-se o acelerômetro em repouso para se fazer medidas amostrais do erro do processo e calculando o desvio-padrão das medidas para se montar a matriz. Já a matriz  $R$  é obtida a partir dos valores de precisão do GPS, encontrado no datasheet.

O estado é referente aos dados de posição e velocidade que se deseja obter, e o equacionamento destas variáveis é feito através da seguinte forma:

$$PosicaoFinal = PosicaoInicial + V\Delta t + \frac{1}{2}a\Delta t^2 \quad (4.2)$$

$$VelocidadeFinal = VelocidadeInicial + a\Delta t \quad (4.3)$$

Escrevendo as equações 4.2 e 4.3 na forma matricial temos algo parecido com a equação 4.1:

$$\begin{bmatrix} Pk \\ Vk \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} Pk-1 \\ Vk-1 \end{bmatrix} + \begin{bmatrix} 0.5\Delta t^2 \\ \Delta t \end{bmatrix} \begin{bmatrix} a \end{bmatrix}$$

Que corresponde a:

$$x_k = F x_{k-1} + B u \quad (4.4)$$

Com isso temos os parâmetros essenciais para o filtro de Kalman, o qual os utiliza para fazer todos os cálculos de predição e atualização: predições de estado, predição da covariância, resíduo da medição, resíduo da covariância, ganho ótimo de Kalman, estado atualizado e covariância estimada.

Na Figura 36 é mostrado o diagrama da fusão dos sensores, evidenciando os processos descritos acima.

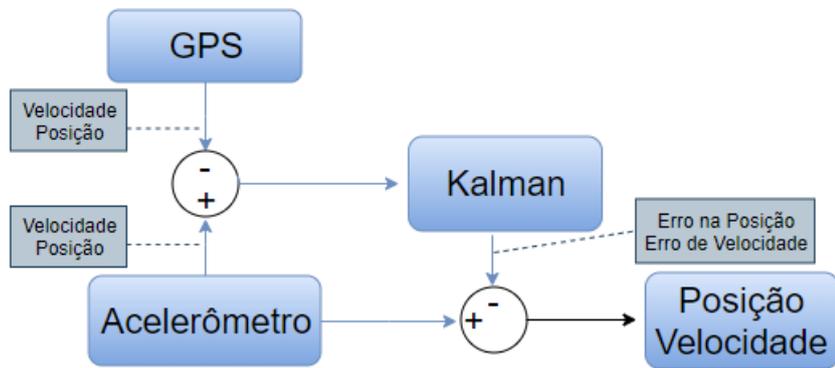


Figura 36 – Esquemático do processo de fusão dos sensores GPS e acelerômetro.

## 5 Metodologia

A metodologia adotada para o desenvolvimento deste trabalho dividiu a realização do projeto em 5 etapas:

- Levantamento de variáveis e requisitos
- Tomada de decisões
- Desenvolvimento do protótipo
- Testes dos subsistemas
- Validação do projeto

A figura 37 apresenta um fluxograma que detalha cada parte do processo de desenvolvimento.

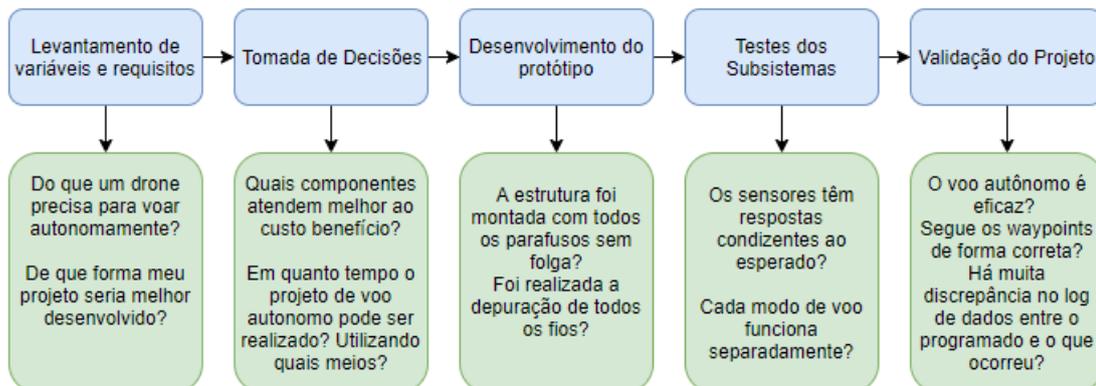


Figura 37 – Etapas da metodologia e suas características

### 5.1 Levantamento de variáveis e requisitos

Esta etapa consistiu primeiramente na realização de pesquisas bibliográficas e, posteriormente, no levantamento de requisitos necessários para que um drone seja autônomo. A partir disso, foram definidas as funcionalidades do protótipo, quais variáveis precisavam ser controladas para um funcionamento eficaz do veículo aéreo, quais seriam os problemas a serem resolvidos no projeto do sistema e, a partir dessa base, foi criado o escopo principal do projeto

Nessa fase já foi possível identificar miniprojetos dentro do projeto global, assim sendo, houve a necessidade de divisão deste em subsistemas menores a serem validados.

Essa divisão possibilitou uma visão mais específica de cada parte do protótipo, o que auxiliou de forma bastante eficiente na resolução de problemas, nos testes e na visão global do funcionamento do protótipo.

## 5.2 Tomada de Decisões

Com a estrutura do projeto já feita e dividida em pontos principais, foi necessário decidir quais componentes (sensores, placa controladora, frame, tipo de energia) se adaptariam melhor aos requisitos propostos e como os problemas principais seriam resolvidos.

Seguindo o objetivo do projeto, a escolha dos componentes obedeceu aos requisitos de qualidade e preço.

Outra decisão tomada é referente à placa controladora de voo, que inicialmente foi pensada em ser toda produzida na IDE do Arduino em linguagem C++, mas, com a finalidade de alcançar o objetivo proposto em tempo hábil, foi alterada para uma placa com códigos já existentes.

Neste sentido, optou-se por realizar uma análise detalhada do algoritmo de voo da placa escolhida, o que servirá de base para uma posterior aplicação na ideia inicial.

## 5.3 Desenvolvimento do Protótipo

Como citado anteriormente, na primeira etapa desta metodologia, para facilitar a solução dos problemas, o projeto foi dividido em subsistemas. Estes são:

- **Hardware:** Este módulo consiste na montagem e configuração de componentes (frame, controladores de velocidade, motores, placa de distribuição de energia, bateria e hélices) de acordo com os requisitos do projeto.
- **Sensores e calibração:** Nesta etapa, todos os sensores essenciais para o voo autônomo são configurados (gps, barômetro, acelerômetro, giroscópio) e calibrados para o voo.
- **Software:** Esse é o subsistema que demanda mais detalhamento, pois é nele que há a análise dos algoritmos de voo em si. Aqui os modos de voo são analisados detalhadamente.

## 5.4 Testes dos Subsistemas

Os testes do subsistema Hardware são feitos utilizando o serial monitor do software Mission Planner para fazer a depuração de dados. Este software já possui a funcionalidade

de testes, então, definindo o componente a ser testado, ele imprime os valores em tela e é possível ver se estão sendo apresentados na forma e no tempo correto.

Já a validação de cada modo de voo é feita a partir de log de dados do voo. Cada vez que um voo é realizado, a placa Ardupilot faz um registro de todos os dados deste. Então, após o voo ser realizado, é possível baixar um registro de log de dados do voo, que é uma ferramenta do Mission Planner, que já apresenta de forma gráfica qual era a variável desejada, qual foi a variável alcançada e, a partir destes dados, é possível fazer uma estimativa de erro do veículo aéreo.

## 5.5 Validação do projeto

A validação do projeto acontece quando este já é capaz de realizar os modos de voo AltHold e Loiter e, a partir da incorporação destes, o modo autônomo é realizado. Deste modo, coordenadas geográficas (waypoints) são colocadas no software Mission Planner, e o veículo deve seguir até elas.



## 6 Resultados

### 6.1 Validação dos Motores

O bom funcionamento do veículo aéreo depende principalmente da força dos seus motores e, especialmente, dos efeitos de rotação dos mesmos, para manutenção do seu voo, sendo assim, a escolha de seus motores é um ponto crucial para que este seja capaz de proporcionar estabilidade no voo.

Portanto, é conveniente que o drone seja capacitado para levantar seu próprio peso e trabalhar com uma certa “folga” para manter um voo eficaz e não consumir a bateria rapidamente, então é necessário saber a propulsão necessária para realizar o voo. O peso do drone será de 1 kilo e duzentas e seis gramas, obtido pelo peso de todos os componentes em conjunto, presentes na Tabela 2 da seção 2.4.6, e validado em uma balança.

#### 6.1.1 Teste de Empuxo

Como apresentado na Seção Motores e Hélices, o empuxo necessário para realização do voo pode ser calculado, a partir da Fórmula 2.1, para cada motor. A Fórmula 2.1 da seção 2.4.6 depende do peso total da estrutura sem os motores, 1 kilo e 14 gramas, e do número de motores presentes na estrutura, no caso, quatro. Então, o empuxo necessário para cada motor é calculado a partir do dobro do peso do drone sem motores, 2 kilos e 28 gramas, dividido pelo número de motores, que fica 507 gramas por motor, na teoria.

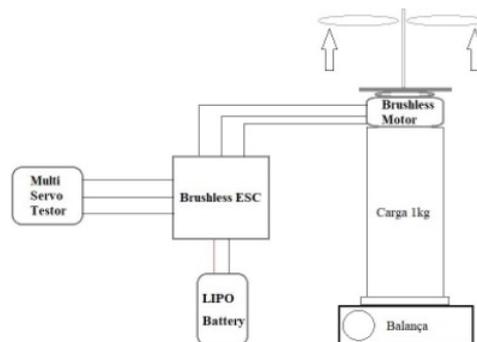


Figura 38 – Esquemático do teste de empuxo

A partir de testes de empuxo, é possível determinar experimentalmente qual o empuxo de cada motor e realizar uma comparação com a teoria. O teste consiste na montagem de uma estrutura, conforme mostrado na figura 38, e é um teste de empuxo

estático, pois a medição é feita com o conjunto parado. Mesmo não sendo uma medida precisa, esses testes fornecem uma aproximação da capacidade de cada motor.

Os quatro motores foram testados individualmente, estando estes alimentados por uma bateria de 5200 mAh, e a corrente gerada foi controlada por um potenciômetro.

Ao acionar o motor, uma força vertical é produzida pela rotação das hélices e essa força tende a levantar o sistema de carga de 1 kg. Com isso, é possível medir o empuxo a partir da massa apresentada na balança que é o quanto o sistema pesa menos o quanto conseguiu levantar da carga utilizada. O teste foi realizado com uma garrafa contendo 1 litro de água, que equivale a 1 kilo, presente na Figura 39. Após o motor ligado, quando acelerado pelo potenciômetro, a hélice tentou levantar o sistema todo, marcando a força de empuxo, ou força de arraste do motor.



Figura 39 – Teste de empuxo

Os resultados obtidos no teste de empuxo estão presentes na Tabela 4.

Tabela 4 – Resultados dos testes de empuxo

Motores	Empuxo Médio
Motor Horário 2	548g
Motor Horário 4	545g
Motor Anti-Horário 1	538g
Motor Anti-Horário 3	542g

Na prática, temos que um modelo para voar não deve pesar mais que 3 vezes o

empuxo estático e para voar bem este valor deve ficar em torno de 2 vezes (E-VOO, 2016). Realizando uma comparação entre os valores presentes na Tabela 4 e o valor teórico de empuxo necessário para cada motor, 507 gramas por motor, constatou-se que o conjunto motor-hélice escolhido aplica-se de forma satisfatória ao projeto, pois nos resultados dos testes os motores possuem 544g de empuxo médio. Percebeu-se que a escolha da estrutura mecânica também foi acertada, pois analisando a força resultante dos atuadores, o sistema terá uma maior facilidade de se estabilizar.

## 6.2 Desenvolvimento do Protótipo

A construção do protótipo deve ser sistemática e testada a todo momento para que não haja erros quando a estrutura estiver montada, sendo assim, para a montagem foram seguidos os passos:

- Solda do conector da bateria na placa de alimentação: o PowerModule é o dispositivo responsável por conectar a bateria à placa central, ou seja, é de suma importância para o protótipo pois é neste ponto que entra a energia que alimentará todo o sistema.
- Soldas dos ESC's na placa de alimentação: é essencial que as soldas sejam bem feitas, pois toda fonte de corrente/tensão da bateria para os motores passará nestes fios. São quatro dispositivos, um para cada motor, que possuem dois cabos referentes à alimentação, um preto e um vermelho, que devem ser soldados na placa nos pólos negativos (-) e positivos (+) respectivamente, como mostrado na figura 40.

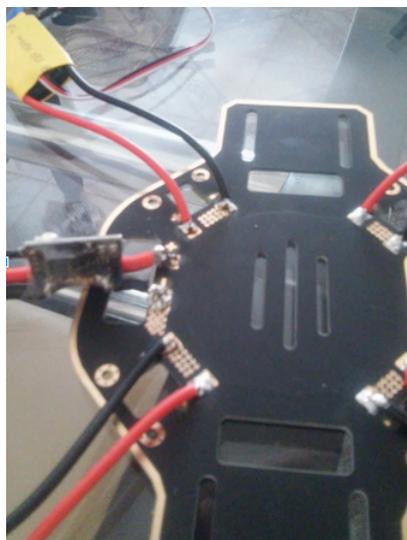


Figura 40 – Controladores de velocidade soldados à placa de distribuição

- Teste de continuidade com multímetro: verifica se a solda dos ESC's e do conector da bateria foram feitas corretamente e o fio está conduzindo do início ao fim.
- Teste de alimentação: ligar a bateria ao conector da placa de alimentação, nesse momento os dispositivos serão energizados, devendo ter 11.1V na placa e 5V em uma saída do ESC chamada BEC (battery eliminator circuit), apresentado na figura 41.

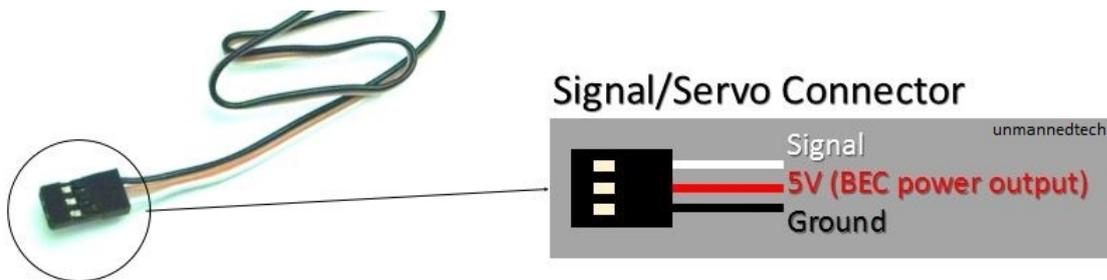


Figura 41 – Saída BEC do ESC

- Fixação dos braços na placa principal (placa de alimentação)
- Fixação dos motores com parafusos nos braços, como apresentado na figura 42.



Figura 42 – Braços e motores parafusados à estrutura

- Montagem dos ESC's nos braços: nesta parte acontece a fixação dos ESC's aos braços do frame por abraçadeiras.
- Disposição e fixação dos dispositivos necessários para o controle na placa central: esta etapa é decisiva na montagem da aeronave. O Ardupilot, o receptor de sinais e

o receptor de telemetria do drone devem ser colocados na placa de forma uniforme, ou seja, é necessário que eles fiquem em uma distribuição onde o peso de todos fique uniformemente distribuído, deixando o centro de massa do drone no centro de sua estrutura, não desnivelando assim o voo do veículo.

- ATENÇÃO: antes da fixação do Ardupilot ao frame, é necessário que o barômetro não esteja recebendo luz ou vibrações, para isso é válido utilizar espumas não-condutoras e fitas isolantes na capa que protege a placa para evitar essas variáveis indesejadas.
- Fixação da haste do GPS e Bússola na placa superior do drone: a haste é item imprescindível para o bom funcionamento da bússola, pois ela afasta o dispositivo dos outros para que não haja interferência magnética nesta.
- Fixação da placa superior do drone na estrutura com parafusos
- Ligação entre GPS e Ardupilot
- Estrutura finalizada
  - ATENÇÃO: é recomendado que as hélices só sejam colocadas no drone momentos antes de voar, ou seja, quando este estiver no seu local de decolagem, para evitar acidentes durante seu transporte.

A figura 43 apresenta um esquemático da montagem do hardware do protótipo.

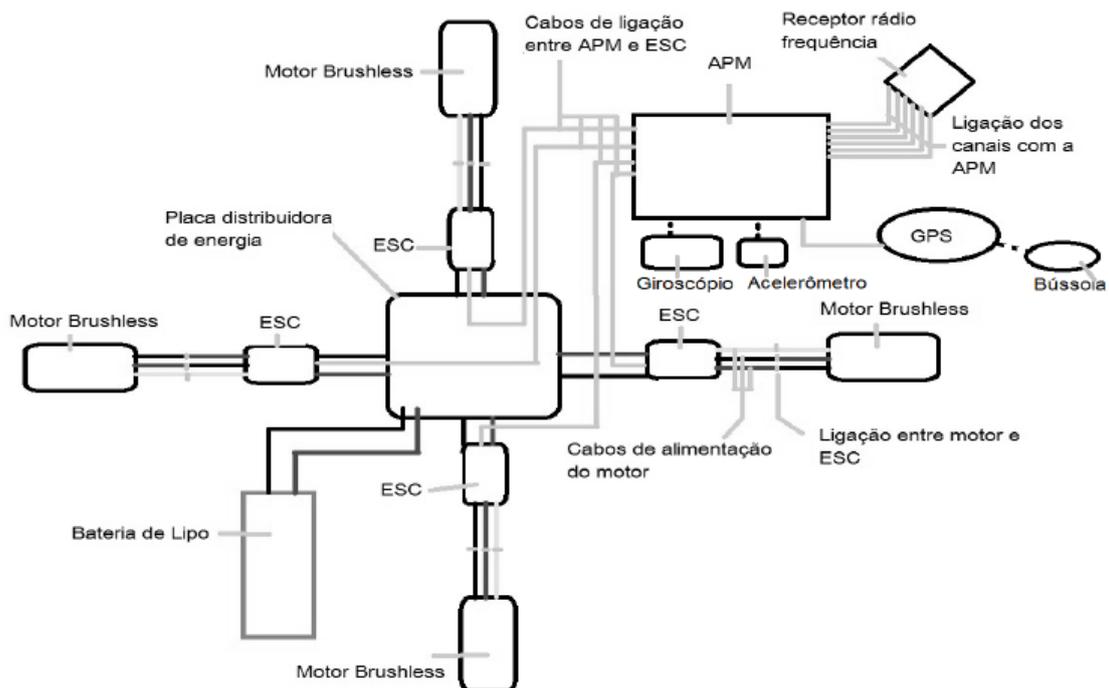


Figura 43 – Esquemático de montagem do protótipo

A Figura 44 apresenta a montagem completa do quadricóptero com vista lateral.



Figura 44 – Vista lateral do veículo aéreo

A figura 45 apresenta uma visão superior do veículo.



Figura 45 – Vista superior do veículo aéreo

A figura 46 apresenta a montagem final do drone com as hélices.



Figura 46 – Drone completo após montagem

## 6.3 Calibração dos Sensores e Dispositivos

### 6.3.1 Calibração do rádio controle

A calibração do rádio controle é de suma importância para o voo, é neste procedimento que a controladora de voo relaciona os movimentos do stick do rádio controle com o quão os motores vão ser acelerados. O rádio controle envia informação de movimentos dos sticks e chaves e o receptor que está conectado à controladora recebe e envia os dados para esta.

A calibração do rádio foi feita pelo software Mission Planner, nesse caso, feita em Configuração Inicial > Hardware Obrigatório > Calibração do Rádio. Em seguida, observamos como estão as barras do Roll, Pitch, Yaw, Throttle e as chaves. Movendo os sticks e as chaves no controle, podemos visualizar as barras movimentando-se de acordo com o comando aplicado, a figura 47 apresenta as barras e em vermelho são os limites máximo e mínimo que o software já recolheu de acordo com os movimentos. O sticks e as chaves devem ser movimentados em todas as direções possíveis.



Figura 47 – Calibração do Rádio com limites dos sticks e chaves em vermelho.

A figura 48 apresenta o rádio controle com seus elementos em destaque.



Figura 48 – Rádio controle e seus elementos

A figura 48 apresenta:

- Em vermelho, encontra-se o stick do controle responsável pela aceleração dos motores e pelo rumo do drone, ou seja, pelo movimento de yaw.
- Em verde, encontra-se o stick do controle responsável pelas inclinações do drone, tanto a inclinação frontal, movimento pitch, quanto a inclinação lateral, movimento roll.
- Em azul, encontra-se a chave do controle responsável por selecionar os modos de voo da aeronave, ela possui três posições, uma para Stabilize, uma para Loiter e uma para Auto.
- Em amarelo e branco, encontram-se os os trims.

#### 6.3.1.1 Trim

Os trims são ajustes finos dos sticks, ou seja, o trim amarelo é responsável por dar um ajuste fino para o stick de aceleração e leme e o trim branco para as inclinações. Esse ajuste é feito durante o voo e muda a referência do neutro do rádio, sendo assim, se o drone está voando um pouco inclinado, durante o próprio voo, o piloto ajusta no trim (movendo o botão para cima ou para baixo), vê a mudança na posição do drone, ou seja, o quão a referência deve ser mudada para o voo ficar “reto” e estável e salva-o.

Para realizar essa configuração, é necessário que uma chave seja colocada com SAVE TRIM.

Como fazê-lo:

A cada movimentação do botão de ajuste do trim, é feito um som. Quando mais alto o som, mais o movimento está se distanciando do centro, ou seja, maior está sendo a mudança na inclinação/aceleração/leme do seu drone.

Ao notar que seu drone está com voo estável, pause-o e mexa na chave do Save Trim. Recoloque todos os sticks e chaves no centro.

Como funciona o algoritmo da calibração do rádio controle em código:

Como foi apresentado na seção 2.4.2, a potência do motor é relacionada à modulação da largura do pulso do sinal de energia (PWM), dessa forma, a adequação dos sticks à rotação dos motores significa que o máximo/mínimo de movimentação dos sticks será o máximo/mínimo de PWM enviado ao motor, como apresenta a figura 49.

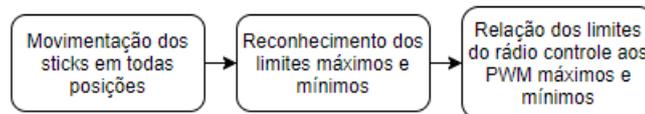


Figura 49 – Aquisição dos limites do rádio controle

O trim funciona da mesma forma, modificando os sinais PWM's enviados ao motor. O movimento do botão do trim para o código, significa um incremento ou um decremento no PWM que será enviado. Para ter um voo estável, parâmetros pré-definidos pela controladora e pela calibração inicial são utilizados, mas nem sempre realiza a estabilização desejada, o trim entra em ação, então, para modificá-los incrementando-os ou decrementando-os. A figura 50 apresenta um fluxograma do algoritmo responsável pelo trim.

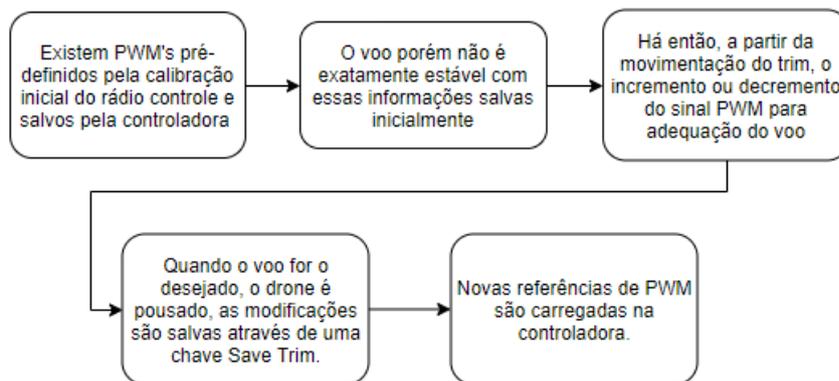


Figura 50 – Algoritmo do Trim responsável pela mudança dos sinais PWM

### 6.3.1.2 Calibração da Unidade Inercial

O drone necessita saber exatamente a posição e direção de seus eixos de orientação para que mantenha a estabilidade e para isso é indispensável a calibração do giroscópio e do acelerômetro.

No Mission Planner a calibração é realizada em Configuração Inicial > Hardware Obrigatório > Calibração Acelerômetro.

O software solicitará que você coloque o veículo em várias posições para reconhecê-las e adequar seus parâmetros a elas. Pressione qualquer tecla para indicar que o piloto automático está em posição e depois prossiga para a próxima orientação.

As posições para calibração são: nivelado, que é a mais importante, pois esta será a referência para o voo se autonivelar, virado para o lado direito, virado para o lado esquerdo, nariz para baixo, nariz para cima e de cabeça para baixo que fornecerão os dados para as manobras do drone.

### 6.3.1.3 Calibração da Bússola (Compass)

Esta calibração foi realizada a partir do software Mission Planner e é conhecida como calibração offboard ou ao vivo.

É realizada em Configuração Inicial > Hardware Obrigatório > Compass. Selecione a opção de Compass externo, pois este dispositivo está acoplado ao GPS.

Uma janela, presente na figura 51, aparecerá para o acompanhamento da calibração da bússola em tempo real.

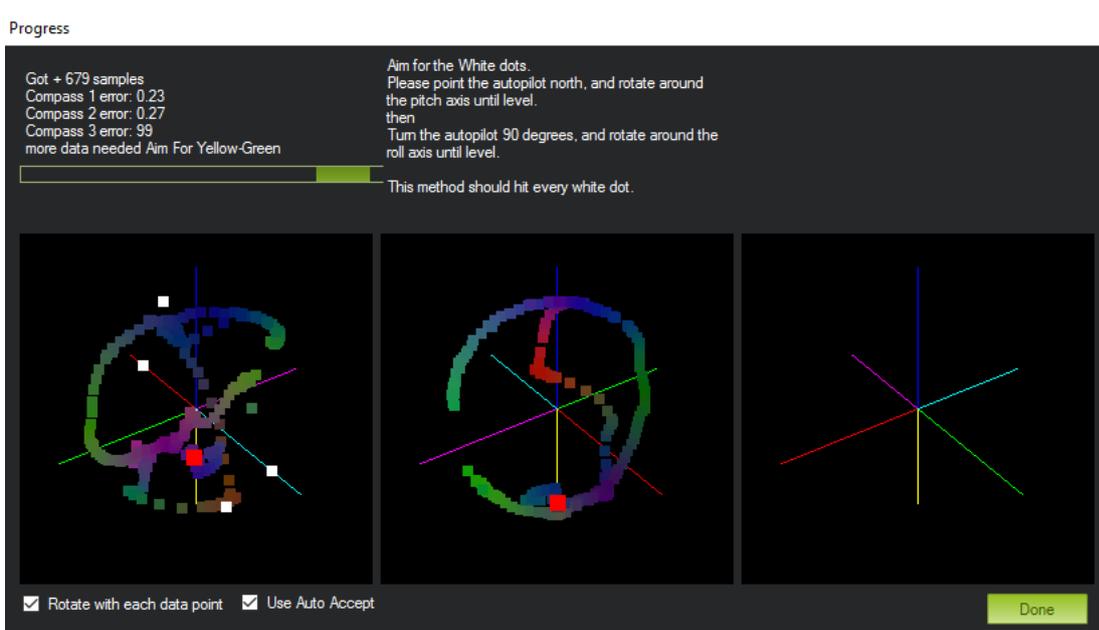


Figura 51 – Calibração da Bússola

O objetivo é recolher os dados de posição da bússola, então o drone deve ser movimentado em várias posições até que os pontos em branco sejam preenchidos. Quando a controladora recolher todos os dados necessários a calibração termina automaticamente.

#### 6.3.1.4 Calibração dos ESC's

Os ESC's precisam ser calibrados para reconhecerem quais são os limites máximos e mínimos de PWM's, pois ele é o responsável por "girar" os motores na velocidade solicitada pelo piloto automático. Portanto, este procedimento só pode ser feito após a calibração do rádio controle por causa da identificação desses sinais.

**ATENÇÃO:** para evitar qualquer tipo de acidente é necessário que essa calibração seja feita sem as hélices.

Passos:

- 1 - Mantenha o drone sem nenhuma fonte de alimentação
- 2 - Ligue o rádio controle, que é o transmissor, e coloque o acelerador no máximo.
- 3 - Ligue a bateria ao drone. Neste momento, a controladora de voo reconhece que deve entrar no modo de calibração dos ESC's.
- 4 - Com o acelerador no máximo ainda, desconecte e reconecte a bateria.
- 5 - A APM piscará leds vermelhos, aguarde os ESC' emitirem 3 bipes (que é o número de células presentes na bateria).
- 6 - Logo em seguida, dois bipes serão dados pelos ESC's que significam que eles reconheceram que o acelerador está no máximo e este será o limite máximo do PWM.
- 7 - Agora é a vez do reconhecimento do limite mínimo do ESC, então, o acelerador deve ser colocado em sua posição mínima.
- 8 - A partir disso, os ESC's emitiram um longo bipe que significa que o dado de PWM mínimo foi recolhido.
- 9 - Nesse momento, eleve o acelerador e os motores devem girar. Calibração finalizada.

### 6.3.2 Modos de Voo

#### 6.3.2.1 Independentes de GPS

Há modos de voo, como o Stabilize e o AltHold, que não dependem do GPS para funcionar, pois não precisam saber qual a localização geográfica do veículo aéreo. No Stabilize, o drone precisa realizar autonivelamento e autoestabilização, e o modo AltHold necessita saber a altitude do veículo, dependendo apenas do barômetro.

O fluxograma da figura 52 apresenta a arquitetura para modos de voo que não dependem do GPS ou da posição do quadricóptero. É necessário salientar que a arquitetura e a explicação posterior referem-se apenas ao modo de voo Stabilize, mas isso é só a critério da explicação, pois tudo se aplica ao modo de voo AltHold também.

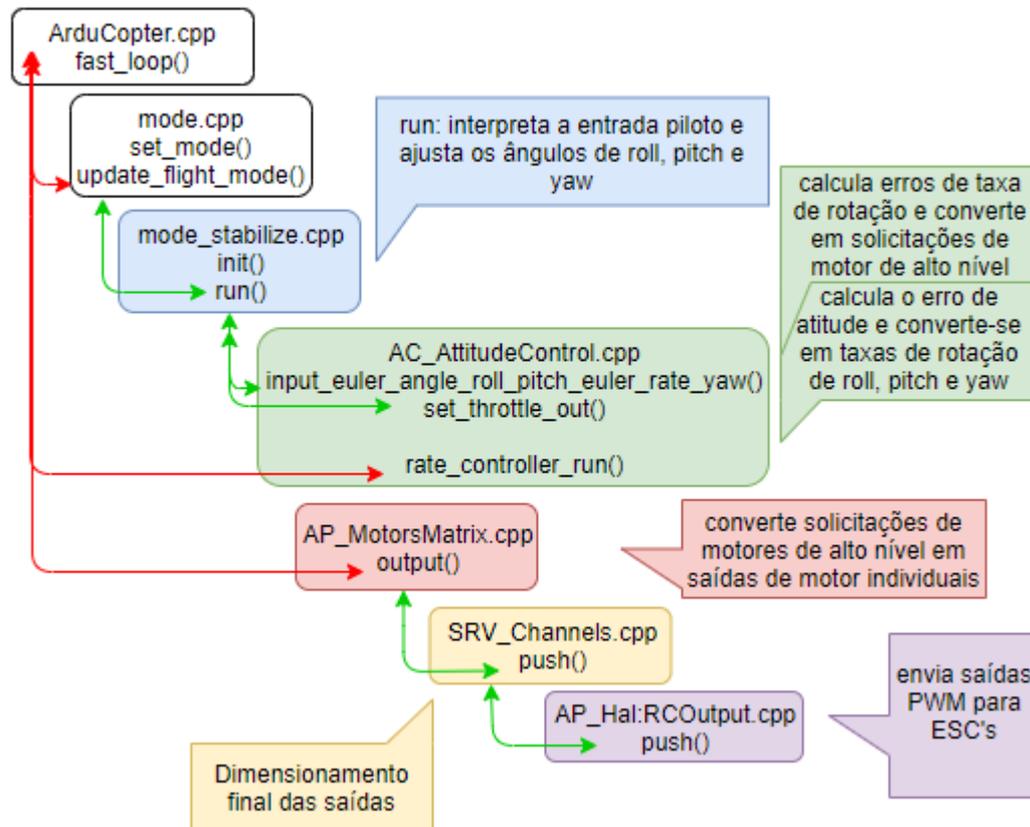


Figura 52 – Arquitetura dos modos Stabilize e AltHold

A figura 52 apresenta o que acontece em cada `fast_loop` (100 Hz):

- A função “`update_flight_mode()`” que pertence à `mode.cpp` é chamada. Esta função verifica o modo de voo do veículo, isto é, verifica a variável “`control_mode`”, que depende da chave seletora do modo de voo do rádio controle. Posteriormente, é chamada a função que possui o modo de voo apropriado, nesse caso o `stabilize_run`. A função `stabilize_run()` pode ser encontrada no arquivo `control_stabilize.cpp`.
- A função `stabilize_run` é responsável por converter a entrada do usuário, (localizada em `g.rc_1.control_in`, `g.rc_2.control_in`, etc) em um ângulo de inclinação, taxa de rotação e taxa de subida. Por exemplo, converte a entrada de roll e pitch do usuário para ângulos de inclinação, a entrada de yaw é convertida em uma taxa de rotação em graus por segundo e a entrada de aceleração é convertida em uma taxa de subida em centímetros por segundo.

- a última coisa que a função `stabilize_run` deve fazer é passar esses ângulos e taxas desejados para a biblioteca Controle de Atitude, mantida na pasta `AC_AttitudeControl`.
- A biblioteca `AC_AttitudeControl` fornece 5 maneiras possíveis de controlar a atitude do veículo, sendo as mais comuns 3 descritas abaixo.

1- `angle_ef_roll_pitch_rate_ef_yaw()`: essa situação aceita ângulos diretos para o roll e para o pitch, mas para o yaw aceita apenas uma taxa de mudança. Por exemplo, receber entradas de `roll = -1000`, `pitch = -1500` e `yaw = 500` significa inclinar o veículo para 10 graus, avançar para 15 graus e girar à direita em 5 graus/segundo, tendo como referência a Terra, ou seja, os graus são de acordo com o norte desta.

2- `angle_ef_roll_pitch_yaw()`: essa maneira aceita ângulos diretos para roll, pitch e yaw. Como é possível perceber, a entrada do yaw não significará mais uma taxa e sim um ângulo direto, diferente da situação anterior. Portanto, a entrada `yaw = 500` significará um giro de 5 graus a leste do norte do veículo, tendo como referência a Terra, ou seja, os graus são de acordo com o norte desta.

3- `rate_bf_roll_pitch_yaw()`: nesta forma todas as entradas aceitas são taxas em graus por segundo, tanto para pitch, como para yaw e roll. Porém, a referência neste método não é mais a Terra, passando a ser o próprio frame, ou seja, a própria estrutura do veículo. Por exemplo, fornecendo para esta função `roll = -1000`, `pitch = -1500`, `yaw = 500` o veículo rodará para a esquerda a 10 graus por segundo, inclinará a frente para baixos em 15 graus por segundo e girará em torno do seu próprio eixo z a 5 graus por segundo.

Após qualquer chamada para essas funções, a função `rate_controller_run()` da `AC_AttitudeControl.cpp` é chamada. Ela converte a saída dos métodos listados acima em entradas roll, pitch e yaw que são enviadas para a biblioteca `AP_Motors` através das funções `set_roll`, `set_pitch`, `set_yaw` e `set_throttle`.

- A biblioteca `AP_Motors` contém o código `AP_MotorsMatrix`. Esse código é responsável por converter os valores de roll, pitch, yaw e throttle, recebidos da biblioteca `AC_AttitudeControl` em saídas absolutas para o motor, valores PWM. Portanto, as bibliotecas de nível mais alto usariam essas funções:

1- `set_roll()`, `set_pitch()`, `set_yaw()` : aceita valores de roll, pitch e yaw no intervalo de -4500 até 4500. Estes não são ângulos desejados ou mesmo taxas, mas apenas um valor. Por exemplo, `set_roll(-4500)` significaria rolar para a esquerda o mais rápido possível.

2- `set_throttle()`: aceita um valor de aceleração absoluta no intervalo de 0 a 1000. Onde, zero equivale a motores desligados e 1000 equivale a motores com aceleração máxima.

3- A função “output\_armed” é responsável por implementar a conversão desses valores de roll, pitch, yaw e throttle em saídas pwm. Essa conversão geralmente inclui a implementação de medidas de estabilidade, que lida com priorização de um eixo de controle sobre outro quando os pedidos de entrada estão fora dos limites físicos do frame. Isto é, não é possível realizar aceleração máxima e rotação máxima em todos os motores de um quadricóptero, pois na execução de alguns movimentos, como movimento roll, alguns motores devem ser menos acelerados do que outros. Na parte inferior da função “output\_armed” existe uma chamada para o `hal.rcout->write()` que passa os valores desejados do pwm para a camada AP\_HAL.

4- Em particular, a função `hal.rc_out_write()` fará com que o PWM especificado, recebido da classe AP\_Motors, apareça no pino pwm apropriado para a placa, ou seja, apenas um arranjo para o hardware.

### 6.3.2.2 Dependentes de GPS

Modos de voo que necessitam saber a localização geográfica do veículo necessitam das coordenadas geográficas advindas do GPS. Dois exemplos são: o Loiter, que é em suma o controle de posição do veículo e para isso precisa reconhecer em qual posição está, e o Auto, que é a incorporação de vários modos de voo para realização de uma missão.

O fluxograma da figura 53 apresenta a arquitetura para modos de voo que dependem da posição do quadricóptero. É necessário salientar que a arquitetura e a explicação posterior referem-se apenas ao modo de voo Auto, mas isso é só a critério da explicação, pois tudo se aplica ao modo de voo Loiter também.

Como é possível perceber, a arquitetura dos modos de voo que dependem da posição para funcionar é extremamente parecida com a dos independentes de GPS, diferenciando-se em apenas dois pontos:

- AC\_PosControl: permite o controle de posição 3D do veículo. Normalmente apenas os métodos mais simples de controle do eixo Z, altitude, desta biblioteca são utilizados. Isto acontece porque os modos de voo mais complicados de posição 3D, como o Loiter, fazem uso da biblioteca AC\_WPNav. Portanto, alguns métodos comumente usados dessa biblioteca são:
  - `set_alt_target_from_climb_rate()`: aceita uma taxa de subida em cm/s e atualiza uma altitude absoluta de referência.
  - `set_pos_target()`: aceita um vetor de posição 3D que é um deslocamento, em cm, do ponto de partida do Drone (Home).
- AC\_WPNav.cpp: é responsável por calcular a trajetória do Drone em um voo autônomo e atualizar os alvos do controlador de posição. Todos as funções da

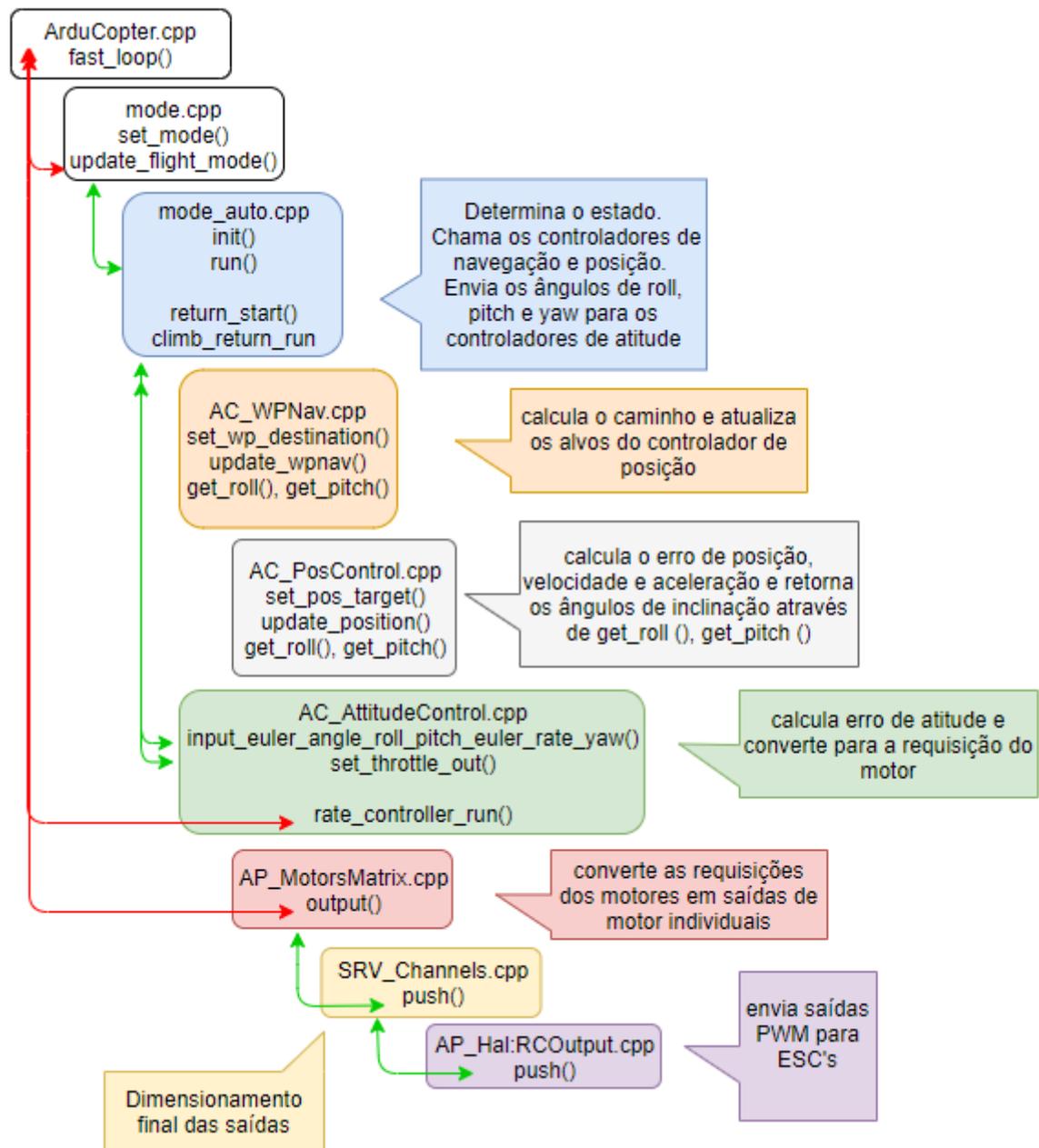


Figura 53 – Arquitetura dos modos Loiter e Auto

“AC\_WPNav” são muito importantes para o voo autônomo e a seguir são descritas algumas delas:

- `set_wp_origin_and_destination()`: Esta função define um waypoint de origem através do vetor “`Vector3f&origin`” e um waypoint de destino através do vetor “`Vector3f&destination`”. A biblioteca “`Location`” é a responsável por transformar os dados de latitude, longitude e altura em vetores. Além disso a variável booleana “`terrain_alt`” é responsável por definir se as alturas dos waypoints são possíveis, ou seja, estão a uma altura superior ao chão, assinalando verdadeiro. Nesta função é determinado verdadeiro se as alturas “`origin.z`” e “`destination.z`”

são superiores a do solo.

- `get_wp_stopping_point_xy()`: Calcula o ponto de parada com base na posição, velocidade e aceleração no ponto atual.
- `calc_slow_down_distance()`: Calcula a distância antes do waypoint em que o veículo deve começar a desacelerar, supondo que este esteja a toda velocidade.

Se alguma função de “AC\_PosControl” for chamada, o código do modo de voo também deve chamar a função “update\_z\_controller()” da mesma biblioteca. Isso executará os loops PID de controle de posição do eixo Z e enviará um nível de aceleração de baixo nível para a biblioteca "AP\_Motors". Se qualquer função do eixo XY for chamada, a função “update\_xy\_controller ()”, também de “AC\_PosControl”, deverá ser chamada.

## 6.4 Análise de Desempenho a partir de logs de dados

A análise dos logs permite a validação do projeto, sendo possível acompanhar de maneira intuitiva e visual se as variáveis reais de voo estão muito divergentes das esperadas. São analisadas 5 tipos de variáveis: a altitude, as posições no eixo X e Y e as velocidades no eixo X e Y, a fim de validar o controle de altitude e de posição.

Para a análise são utilizados dados dos logs de voo no momento em que o veículo está efetuando um voo autônomo em uma missão simples. Para uma melhor comparação, todos os dados utilizados estão na mesma linha do tempo, mesmo para as diferentes variáveis. Com o intuito de verificar a dispersão entre os dados desejados e os dados medidos também são mostrados gráficos de erro relativo, os dados utilizados para criar estes gráficos foram importados do Mission Planner em um arquivo txt.

### 6.4.1 Posição no Eixo X

O controle de posição no eixo X é analisado a partir da posição desejada nesse eixo (variável “DposX”, contida no vetor “NTUN”) e a posição aferida durante o voo (variável “PosX”, contida no vetor “NTUN”). O gráfico obtido está presente na figura 54.

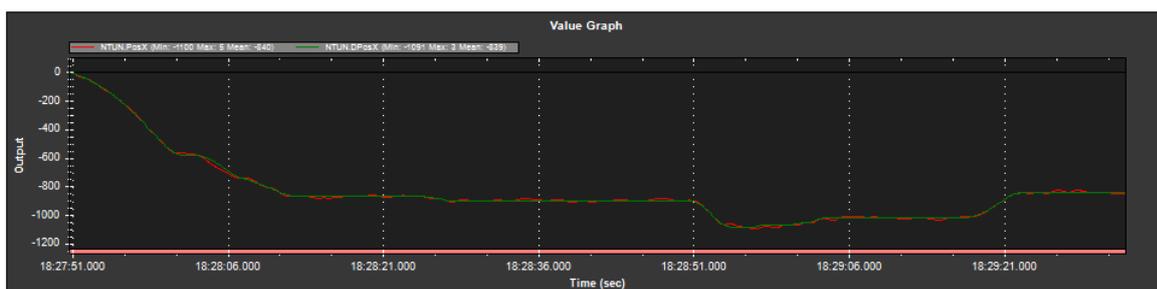


Figura 54 – Gráfico de Posição no Eixo X

A figura 54 apresenta em verde a posição desejada no eixo X e em vermelho a posição real. Sendo assim, existe um erro associado às duas medidas, dessa maneira, um gráfico que apresenta o erro percentual das medidas em relação ao tempo foi gerado e está presente na figura 55.



Figura 55 – Gráfico de Erro Relativo por Tempo das medidas da posição em X

Como é possível perceber nos dois gráficos, não houve divergências consideráveis entre as medidas reais e desejadas.

O eixo em que o erro relativo está presente tem seu limite máximo em 1%, ou seja, não houve erros maiores que esse valor.

#### 6.4.2 Posição no Eixo Y

O controle de posição no eixo Y é analisado a partir da posição desejada neste eixo (variável "DposY", contida no vetor "NTUN") e a posição Y aferida durante o voo (variável "PosY", contida no vetor "NTUN"). O gráfico obtido está presente na figura 56.

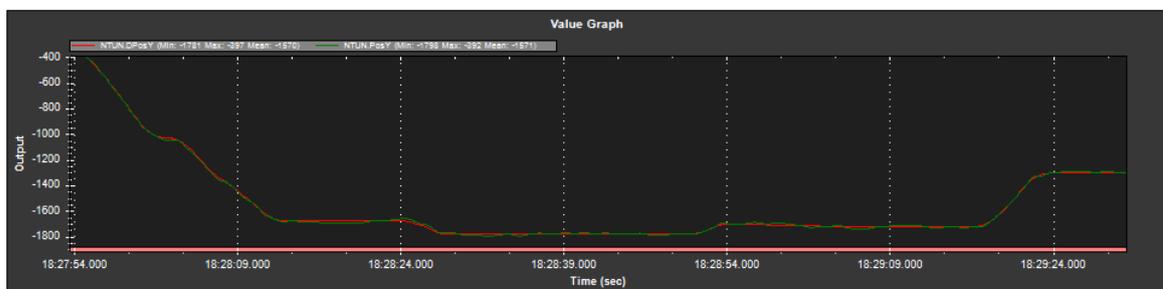


Figura 56 – Gráfico de Posição no Eixo Y

A figura 56 apresenta em vermelho a posição desejada no eixo Y e em verde a real, sendo assim, existe um erro associado às duas medidas. Dessa maneira, um gráfico que

apresenta o erro percentual das medidas em relação ao tempo foi criado e está presente na figura 57.



Figura 57 – Gráfico de Erro Relativo por Tempo das medidas da posição em Y

Como é possível perceber nos dois gráficos, não houve divergências consideráveis entre as medidas reais e desejadas.

O eixo em que o erro relativo está presente tem seu limite máximo em 0,5%, ou seja, não houve erros maiores que esse valor.

### 6.4.3 Altitude

O controle de altitude é verificado a partir dos dados de altitude desejada (variável “Dalt”, contida no vetor CTUN) e a altitude aferida durante o voo (variável “alt”, contida no vetor “CTUN”). O gráfico obtido está presente na Figura 58.

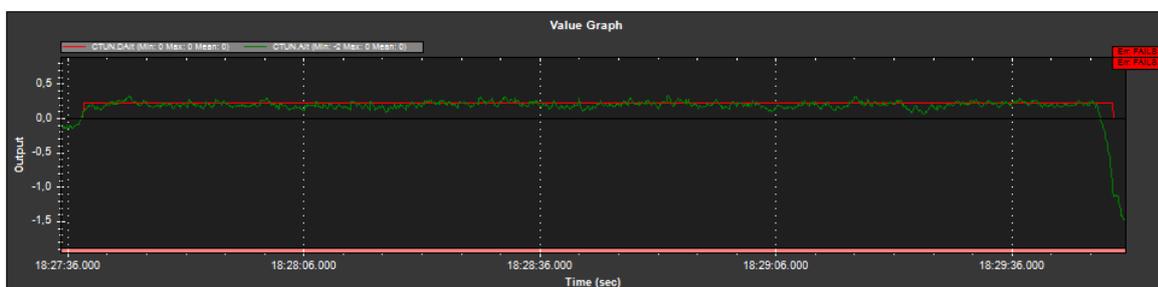


Figura 58 – Gráfico de Altitude

A figura 58 apresenta em vermelho a altitude desejada e em verde a real, sendo assim, existe um erro associado às duas medidas. Dessa maneira, um gráfico que apresenta o erro percentual das medidas em relação ao tempo foi criado e está presente na figura 59.

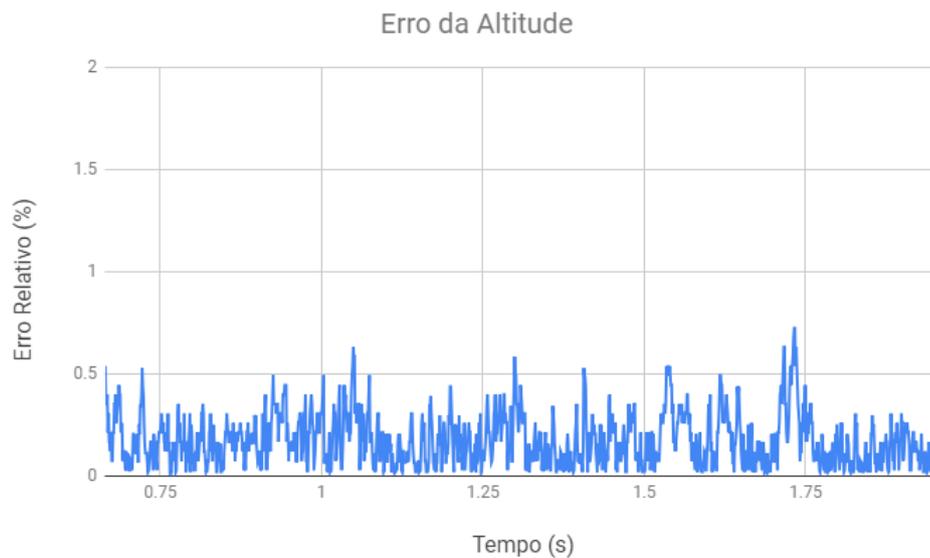


Figura 59 – Gráfico de Erro Relativo por Tempo das medidas de Altitude

Como é possível perceber nos dois gráficos, não houve divergências consideráveis entre as medidas reais e desejadas.

O eixo em que o erro relativo está presente tem seu limite máximo em 2%, ou seja, não houve erros maiores que esse valor.

O controle de altitude é o que apresenta maiores oscilações, isso acontece devido à flutuações no sensor barométrico embarcado no ArduPilot.

#### 6.4.4 Velocidade no eixo X

O controle de velocidade no eixo X é analisado a partir da velocidade desejada nesse eixo (variável “DVelx”, contida no vetor “NTUN”) e a velocidade aferida durante o voo (variável “VelX”, contida no vetor “NTUN”). O gráfico obtido está presente na Figura 60.

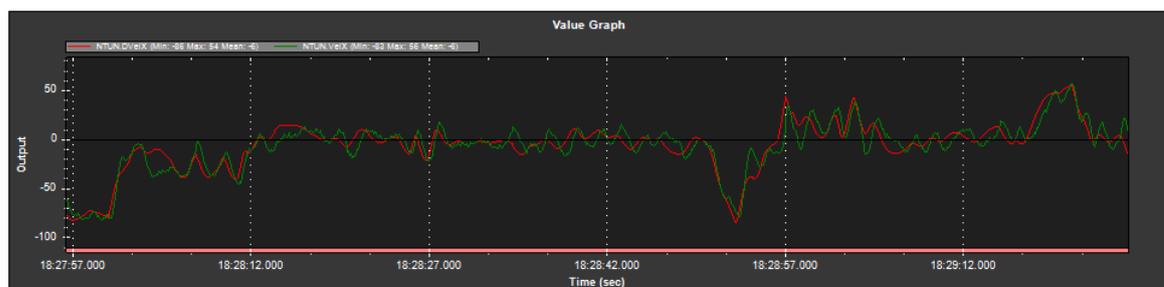


Figura 60 – Gráfico de Velocidade no Eixo X

A figura 60 apresenta em vermelho a velocidade desejada para o eixo X e em verde

a real, sendo assim, existe um erro associado às duas medidas. Dessa maneira, um gráfico que apresenta o erro percentual das medidas em relação ao tempo foi criado e está presente na figura 61.

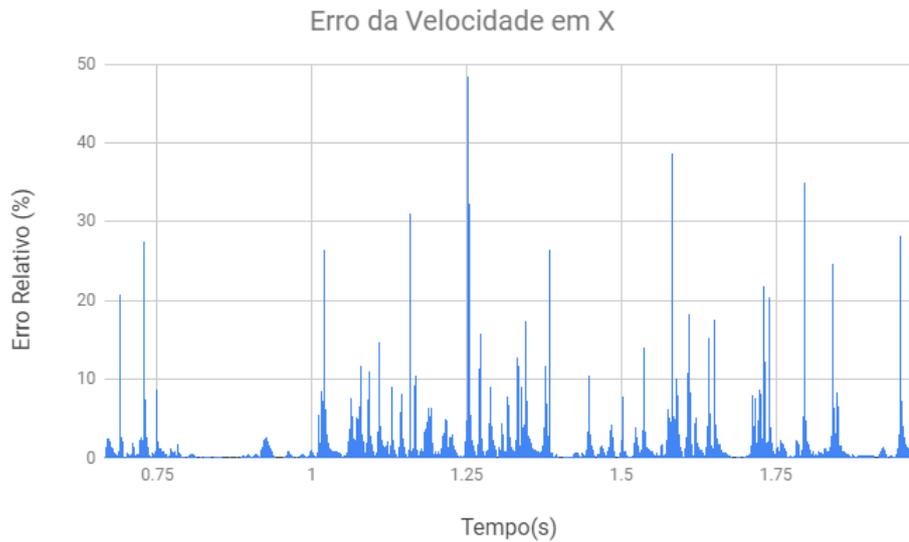


Figura 61 – Gráfico de Erro Relativo por Tempo das medidas de Velocidade em X

O gráfico do erro relativo à velocidade no eixo x apresenta as maiores divergências, mostrando erros próximos à 50%. Porém, o controle se corrige rapidamente e de forma eficaz, trazendo o valor de volta para o desejado. Confirmando que o controle foi bem realizado.

#### 6.4.5 Velocidade no eixo Y

O controle de velocidade no eixo Y é analisado a partir da velocidade desejada nesse eixo (variável "DVely", contida no vetor "NTUN") e a velocidade aferida durante o voo (variável "Vely, contida no vetor "NTUN"). O gráfico obtido está presente na Figura 62.



Figura 62 – Gráfico de Velocidade no Eixo Y

A figura 62 apresenta em vermelho a velocidade desejada para o eixo X e em verde a real, sendo assim, existe um erro associado às duas medidas. Dessa maneira, um gráfico que apresenta o erro percentual das medidas em relação ao tempo foi criado e está presente na figura 63.

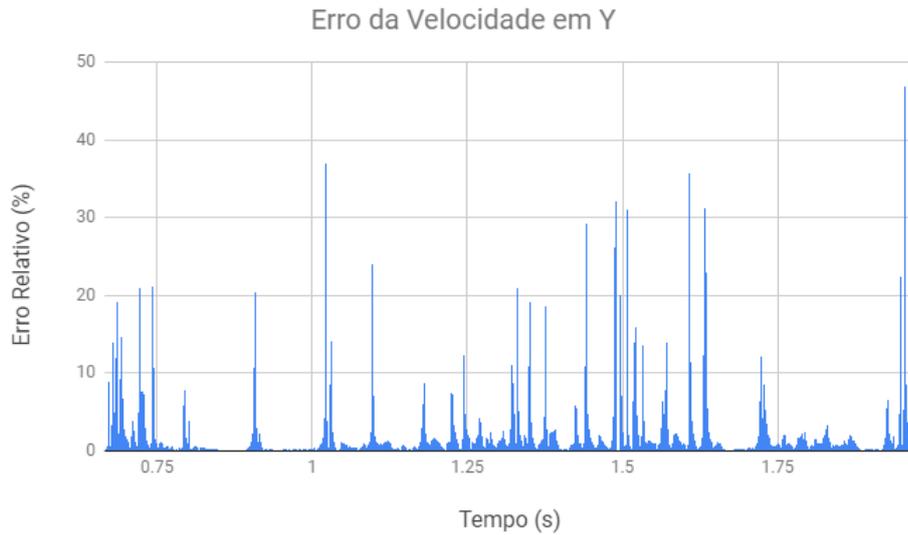


Figura 63 – Gráfico de Erro Relativo por Tempo das medidas de Velocidade em Y

Como é possível perceber dos dois gráficos, mesmo apresentando em algumas situações erros próximos à 40%, o objetivo do voo não foi comprometido e ele realizou-se efetivamente.



## 7 Considerações Finais

O desenvolvimento de um veículo aéreo autônomo envolve inúmeras dificuldades e grande detalhamento, sendo assim, todos os procedimentos relatados durante todo este trabalho devem ser realizados com afinco, atenção e um planejamento bem aprofundado.

Quanto à montagem da estrutura, é necessário que a condutividade dos dispositivos seja checada a cada momento para não acontecer nenhuma falha depois de toda estrutura já está parafusada.

Vale ressaltar que uma boa calibração e aferição de funcionamento dos sensores é essencial, o voo necessita que cada variável seja recebida, tratada e enviada com bastante presteza para realizar voo estável.

No mês de Maio, ocorreu um "fly away" com o primeiro protótipo durante um teste. A telemetria ainda não havia sido implementada no drone e o comportamento deste no voo não era conhecido. O barômetro que veio embarcado no primeiro Ardupilot adquirido oscilava em, mais ou menos, 9 metros e a cada atualização do código, o sistema adequava-se a essa nova altitude que estava errada. Posteriormente, o projeto físico foi reiniciado do zero.

A análise da arquitetura da controladora de voo ArduPilot permitiu todo o desenvolvimento deste projeto. A partir do conhecimento adquirido, foi possível realizar a construção de um protótipo que realiza voos autônomos, além de permitir que projetos futuros sejam baseados neste.

Baseando-se nos testes realizados em campo, assim como na análise dos logs de desempenho do veículo aéreo, é fato que o objetivo foi alcançado com sucesso e a aeronave é capaz de realizar voos autônomos com bastante eficácia, cumprindo em suma todos objetivos planejados em uma missão.

Portanto, cada variável, sensor, componente, fio ou parafuso, deve ser tratado com primazia e atenção. Para o funcionamento deste projeto foram empregados tempo, concentração e, sobretudo, amor.



## 8 Referências Bibliográficas

CAMPELO, Matheus de Oliveira. AUTOMAÇÃO DE VOO. 2018. 40 f. Monografia (Ciências Aeronáuticas)- Universidade do Sul de Santa Catarina, Santa Catarina, 2018. Disponível em: <<https://www.riuni.unisul.br/bitstream/handle/12345/5383/>>. Acesso em: 20 jun. 2018.

UBLOX. NEO-M8 Data Sheet. 2015. 26 f. Disponível em: <[https://www.u-blox.com/sites/default/files/NEO-M8\\_DataSheet\\_\(UBX-13003366\).pdf](https://www.u-blox.com/sites/default/files/NEO-M8_DataSheet_(UBX-13003366).pdf)>. Acesso em: 07 set. 2018.

CORRÁ, João Francisco. MONTAGEM E CONFIGURAÇÃO DE UM QUADROTOR. 2014. 57 f. Monografia (Engenharia Elétrica)- Faculdade de Engenharias, Universidade do Vale do Paraíba, São Paulo, 2014. Disponível em: <<http://biblioteca.univap.br/dados/000014/000014f4.pdf>>. Acesso em: 12 maio 2018.

DIAS, Guilherme de Oliveira. PROJETO DE UM DRONE HÍBRIDO. 2017. 107 f. Monografia (Engenharia Mecânica)- Escola de Engenharia, UNIVERSIDADE FEDERAL FLUMINENSE, Niterói, 2017. Disponível em: <<https://app.uff.br/riuff/bitstream/1/4522/>>. Acesso em: 07 abr. 2018.

AFONSO, Johnnatan Messias Peixoto. FRAMEWORK PARA SISTEMAS DE NAVEGAÇÃO DE VEÍCULOS AÉREOS NÃO TRIPULADOS. 2014. 68 f. Monografia (Bacharelado em Ciência da Computação)- Instituto de Ciências Exatas e Biológicas, Universidade Federal de Ouro Preto, Ouro Preto, 2014.

SANTOS, Thobias Rodolfo. Integração de um Vant. 2015. 56 f. Monografia (Bacharelado em Engenharia Eletrônica)- Faculdade de Engenharias, Universidade do Vale do Paraíba, São Paulo, 2015. Disponível em: <<http://biblioteca.univap.br/dados/000015/00001534.pdf>>. Acesso em: 07 jul. 2018.

ANDRADE, Raúl Duarte Nóbrega. Criação de uma estação de testes de um micro-helicóptero autônomo. 2012. 131 f. Dissertação (Mestrado em Engenharia Informática)- Faculdade de Engenharias, Universidade da Madeira, Ilha da Madeira, 2012.

ANAC. Definição de Piloto automático. Disponível em <[http://www2.anac.gov.br/anac-pedia/por\\_ing/tr2762.htm](http://www2.anac.gov.br/anac-pedia/por_ing/tr2762.htm)> Acesso em 13 mar. 2018

MACHADO, Jonathan. O que é GPS?. Disponível em <<http://https://www.tecmundo.com.br/conexao/215-o-que-e-gps-.htm>> Acesso em 13 mar. 2018

ARDUPILOT. Copter Home. Disponível em <<http://gardupilot.or/copter/>>

COSTA, Sérgio Eduardo Aurélio Pereira da. Controlo e Simulação de um Quadrirotor convencional. 2008. 103 f. Dissertação (Mestrado) - Curso de Engenharia Aeroespacial, Universidade Técnica de Lisboa, Lisboa, 2008. Disponível em:

<<https://fenix.tecnico.ulisboa.pt/downloadFile/395137818652/Tese - Final.pdf>>.

Acesso em: 01 de Setembro 2017

TRINDADE, Rafael Henrique. Estudo de Máquinas Elétricas Não-Convencionais: Motor Brushless DC. 2009. 39 f. TCC (Graduação) - Curso de Engenharia Elétrica, Universidade de São Paulo, São Carlos, 2009.

LEISHMAN, J. G. A History of Helicopter Flight. University of Maryland. Disponível em: <http://terpconnect.umd.edu/~leishman/Aero/history.html>, 2000.

PFEIFER, Erick. Projeto e Controle de um UAV quadrirotor. 2013. 132 f. Dissertação (Mestrado) - Curso de Engenharia de Sistemas, Universidade de São Paulo, São Paulo, 2013. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/3/3139/tde-06072014-223705/pt-br.php>>. Acesso em: 5 de Junho de 2017.

SILVA, Guilherme Marques Amaral. Desenvolvimento e Controlo de um Veículo Autónomo Aéreo Quadrotor. 2009. 177 f. Dissertação (Mestrado) - Curso de Engenharia Elétrica, Instituto Superior de Engenharia do Porto, Porto, 2009.

RESENDE, Daniel de Castro Ribeiro. Controle de estabilidade de aeromodelo tipo quadcopter autômato por lei de controle PID. 2014. 107 f. TCC (Graduação) - Curso de Engenharia Mecatrônica, Pontifícia Universidade Católica de Minas Gerais, Belo Horizonte, 2014.

AVELINO, Leonardo de Lima. PROJETO DE UM CONTROLADOR EMBARCADO PARA UM QUADRIROTOR. 2016. 53 f. TCC (Graduação) - Curso de Engenharia Eletrônica, Universidade de Brasília, Brasília, 2016.

VIEIRA, José Carlos Silva. Plataforma Móvel Aérea QuadRotor. 2011. 104 f. Dissertação (Mestrado) - Curso de Engenharia Eletrônica, Universidade do Minho, Braga, 2011. Disponível em: <<http://intranet.dei.uminho.pt/gdmi/galeria/temas/pdf/48031.pdf>>. Acesso em:

17 de Julho de 2017.

SÁ, Rejane Cavalcante. CONSTRUÇÃO, MODELAGEM DINÂMICA E CONTROLE PID PARA ESTABILIDADE DE UM VEÍCULO AÉREO NÃO TRIPULADO DO TIPO QUADRIROTOR. 2012. 94 f. Dissertação (Mestrado) - Curso de Engenharia de Teleinformática, Universidade Federal do Ceará, Fortaleza, 2012.

BARATO, Bruno . Projeto de um Sistema de Controle para Veículos Aéreos Não Tripulados.

OGATA, Katsuhiko. Engenharia de Controle Moderno. 5. ed. Pearson, 2011.

MOLINA, Marcelo. Dicas essenciais para quem deseja comprar um drone. 2015. Disponível em: <<http://www.molrc.com/?p=94>>. Acesso em: 2 de Novembro de 2017.

CASTILLO, Lozano. Modelling and Control of Mini-Flying Machines. Editora Springer-Verlag, London, 2005.

GARRET, Filipe. O que é drone e para que serve? Tecnologia invade o espaço aéreo. Brasil. 2013. Disponível em <http://www.techtudo.com.br/noticias/noticia/2013/10/o-que-sao-e-para-que-servem-os-drones-tecnologia-invade-o-espaco-aereo.html>. Acesso em: 20 de agosto de 2014.

CALDEIRA, Jonathan. Controlando servos com MPU6050. 2016. Disponível em: <<https://www.zonamaker.com.br/controlando-servos-com-o-mpu6050/>>. Acesso em: 15 de Outubro de 2017.

SIQUEIRA, T.S. Bluetooth - Características, protocolos e funcionamento. Disponível em: <[www.ic.unicamp.br/~ducatte/mo401/1s2006/T2/057642-T.pdf](http://www.ic.unicamp.br/~ducatte/mo401/1s2006/T2/057642-T.pdf)> Acessado em: 27 de Março de 2017.

BR-ARDUINO. Servidor web WiFi com Arduino e ESP8266. Disponível em: <<http://br-arduino.org/2015/07/arduino-servidor-web-wifi-esp8266.html>> Acesso em: 24 de Março de 2017.

LEKSELL, Frederik. Manual of Sensorless Brushless Speed Controller. Disponível em: <<http://frederik.se/wp-content/uploads/2015/12/HW-01-V4.pdf>>. Acesso em: 28 mar. 2017.

ARDUINO E CIA. Sensor GY-80 - Acelerômetro Giroscópio Magnetômetro e Barômetro. Disponível em: <<http://www.arduinoecia.com.br/2014/09/sensor-gy-80-acelerometro-bussola-barometro.html>> Acesso em: 26 de Março de 2017.

MECANICA INDUSTRIAL. O que é controle Eletrônica de Velocidade. Disponível em: <<https://www.mecanicaindustrial.com.br/179-o-que-e-controle-eletronico-de-velocidade/>> Acesso em: 27 de Março de 2017.

PALPITE DIGITAL. Como fazer um drone caseiro com Arduino?. Disponível em: <<https://www.palpitedigital.com/como-fazer-drone-caseiro-arduino/>>. Acesso em: 25 de Março de 2017.

MULTILOGICA. Ardupilot: controlador UAV compatível com arduino Mega. Disponível em: <<https://multilogica-shop.com/ardupilot-controlador-uav-compat%C3%ADvel-com-arduino-mega>>. Acesso em: 25 de Março de 2017.

CAMPOS, Augusto. ESP8266 do jeito simples: conexão ao Arduino, acesso à rede WiFi e resposta a ping. Disponível em: <<http://br-arduino.org/2015/05/esp8266-arduino-facil.html>>. Acesso em: 25 de Março de 2017.

ANDERSON JR, John David. Fundamentals of aerodynamics. Tata McGraw-Hill Education, 2010. Acessado em: 26 de Março de 2017.

NUSSENZVEIG, Herch Moysés. Curso de física básica. E. Blucher, 2002. Acessado em: 26 de Março de 2017.