



UNIVERSIDADE DO VALE DO TAQUARI – UNIVATES
CURSO DE ENGENHARIA DA COMPUTAÇÃO

**USO DE DRONE AUTÔNOMO
PARA AUXÍLIO NA COMPROVAÇÃO DE ALARMES**

Diziano Machado da Conceição

Lajeado, dezembro de 2018

Diziano Machado da Conceição

**USO DE DRONE AUTÔNOMO
PARA AUXÍLIO NA COMPROVAÇÃO DE ALARMES**

Monografia apresentada ao Centro de Ciências Exatas e Tecnológicas da Universidade do Vale do Taquari - Univates, como parte da exigência para a obtenção do título de Bacharel em Engenharia da Computação.

Orientador: Prof. Me. Alexandre Stürmer Wolf

Lajeado, dezembro de 2018

Diziano Machado da Conceição

**USO DE DRONE AUTÔNOMO
PARA AUXÍLIO NA COMPROVAÇÃO DE ALARMES**

A banca examinadora abaixo aprova a Monografia apresentada na disciplina de Trabalho de Conclusão de Curso - Etapa II, do Curso de Graduação em Engenharia da Computação, da Universidade do Vale do Taquari - Univates, como parte da exigência para a obtenção do grau de Bacharel em Engenharia da Computação:

Prof. Me. Alexandre Stürmer Wolf - orientador
Universidade do Vale do Taquari - Univates

Prof. Me. Luis Antônio Schneiders
Universidade do Vale do Taquari - Univates

Prof. Dr. Marcelo de Gomensoro Malheiros
Universidade do Vale do Taquari - Univates

Lajeado, 11 de dezembro de 2018

Dedico este trabalho à minha esposa, por todo o incentivo e paciência durante essa longa jornada de estudos, e aos meus pais, que possibilitaram a minha caminhada até esta importante etapa da minha vida, sempre me apoiando e me incentivando.

AGRADECIMENTOS

Um agradecimento especial ao meu orientador Alexandre Stürmer Wolf, por toda a ajuda e pelas valiosas ideias que contribuíram para a concepção e desenvolvimento deste trabalho. Aos meus professores, os quais admiro e agradeço por todos os ensinamentos, em especial aos que compuseram a banca de avaliação e colaboraram com recomendações para este estudo. Aos meus familiares e amigos, por compreender as minhas ausências e pelo apoio durante o decorrer deste trabalho e de toda a graduação. Agradeço a Deus pelas pessoas que colocou em minha vida, pela saúde que me permitiu chegar até aqui e pela força de vontade que me encorajou nos momentos mais difíceis. Muito obrigado.

RESUMO

Com o advento das tecnologias, estão crescendo as possibilidades de utilização dos Veículos Aéreos Não Tripulados (VANTs), pois tornaram-se mais acessíveis e fáceis de usar, além de fornecerem maior segurança quando aplicados a situações que anteriormente exigiam a utilização de aeronaves tripuladas. À medida que a tecnologia de voos autônomos evoluir, os VANTs poderão realizar tarefas cada vez mais complexas, possibilitando respostas rápidas e precisas a desafios da humanidade, como segurança pública, mudanças climáticas e desastres ambientais. O objetivo deste trabalho consiste na construção de um VANT do tipo *drone* que possua a capacidade de realizar voos autônomos, fornecendo meios de verificar e validar incidentes expostos através do disparo de alarmes, apresentando dados e imagens para o utilizador. O uso de *drones* justifica-se no auxílio do trabalho de equipes de verificação, que passam a ser acionadas somente nos casos onde os incidentes forem confirmados. Para atingir os objetivos, foi realizada uma análise das tecnologias disponíveis no mercado através da pesquisa de referencial teórico, juntamente com o desenvolvimento de um protótipo de *drone* e um software de monitoramento e controle. Ao final do trabalho, concluiu-se que a solução proposta possui um grande potencial, possibilitando ganho de agilidade e praticidade na verificação de alarmes disparados.

Palavras-chave: VANTs, Voos autônomos, *Drones*, Verificação de alarmes.

ABSTRACT

With the advent of technologies, the possibilities of using unmanned aerial vehicles (UAVs) have grown, because they have become more accessible and easier to use, besides provide greater safety when applied to situations that previously required the use of manned aircraft. As autonomous flights technology evolves, UAVs will be able to perform increasingly complex tasks, enabling quick and accurate responses to humanity's challenges, such as public safety, climate change and environmental disasters. The objective of this work is to construct a drone type UAV that has the capacity to perform autonomous flights, providing means of verifying and validating exposed incidents by triggering alarms, presenting data and images to the user. The use of drones is justified in assisting the work of verification teams, which are only triggered in cases where the incidents are confirmed. To achieve the objectives, an analysis of the technologies available in the market was carried out through theoretical reference research, along with the development of a drone prototype and a monitoring and control software. At the end of the work, it was concluded that the proposed solution has a great potential, allowing agility gain and practicality in the verification of triggered alarms.

Keywords: UAVs, Autonomous flights, Drones, Alarm verification.

LISTA DE FIGURAS

Figura 1 – Modos de operações de voo	23
Figura 2 – Eixos de rotação	28
Figura 3 - Giroscópio mecânico	30
Figura 4 – Giroscópio digital.....	31
Figura 5 – Constelação dos satélites GPS	33
Figura 6 – APM 1	35
Figura 7 – APM 2.6	36
Figura 8 – Pixhawk.....	37
Figura 9 – Naza-M Lite e módulo GPS/Bússola	37
Figura 10 – <i>Frame</i> modelo DJI F450	38
Figura 11 – Configuração de <i>frame</i> em “+” ou em “x”	39
Figura 12 – Rádio transmissor.....	39
Figura 13 – Rádio receptor	40
Figura 14 – ESC genérico.....	41
Figura 15 – Diagrama de PWM.....	42
Figura 16 – Características de uma hélice	43
Figura 17 – Hardware de captura e transmissão de vídeo	45
Figura 18 – Pacote MAVLink	46
Figura 19 – Tela principal do Mission Planner	48
Figura 20 – Simulador SITL.....	49
Figura 21 – <i>Drone</i> hexacopter	51
Figura 22 – <i>Drone</i> e seu rádio transmissor.....	53
Figura 23 – Placas desenvolvidas por Melo	54
Figura 24 – <i>Drone</i> desenvolvido por Melo	54

Figura 25 – Decolagem do <i>drone</i>	60
Figura 26 – <i>Drone</i> executando uma missão	61
Figura 27 – Momento em que o <i>drone</i> retorna à coordenada inicial.....	62
Figura 28 – Componentes do protótipo e esquema de conexões.....	63
Figura 29 – <i>Frame</i> S500.....	64
Figura 30 – Motores 2212 980 kV.....	65
Figura 31 – Hélices 1045.....	65
Figura 32 – Bateria LiPo 4200 mAh	66
Figura 33 – ESC OPTO 30 A	66
Figura 34 – Controladora Pixhawk.....	67
Figura 35 – Interruptor de segurança.....	67
Figura 36 – Módulo de Potência	68
Figura 37 – Módulo GPS com bússola.....	68
Figura 38 – Rádio transmissor e receptor.....	69
Figura 39 – <i>PPM Encoder</i> conectado ao rádio receptor.....	69
Figura 40 – Transceptores de telemetria.....	70
Figura 41 – Câmera SJ5000X.....	70
Figura 42 – Receptor e transmissor de vídeo	71
Figura 43 – Adaptador EasyCAP	71
Figura 44 – Componentes soldados à placa de distribuição	73
Figura 45 – Vista superior do <i>drone</i> montado.....	73
Figura 46 – Direção da bússola e da Pixhawk.....	74
Figura 47 – <i>Drone</i> montado	75
Figura 48 – Estação de Controle Terrestre	75
Figura 49 – Telas de calibragem do <i>drone</i>	76
Figura 50 – Voo manual de teste	77
Figura 51 – Balanceamento de hélice sendo verificado	78
Figura 52 – <i>Script</i> de conexão com o <i>drone</i>	79
Figura 53 – Função responsável pela leitura de um arquivo de missão	80
Figura 54 – Função que envia as missões para o <i>drone</i>	80
Figura 55 – Função de decolagem.....	81
Figura 56 – Trecho de inicialização do código.....	81
Figura 57 – Exemplo de execução do <i>script</i>	82
Figura 58 – Arquitetura da GCS.....	82

Figura 59 – Software GCS desenvolvido	83
Figura 60 – Visualização dos pontos de verificação	84
Figura 61 – Representação de envio do <i>drone</i>	84
Figura 62 – <i>Drone</i> em solo	85
Figura 63 – Notificação de alarme disparado	85
Figura 64 – <i>Drone</i> decolando	86
Figura 65 – <i>Drone</i> sobrevoando o local do alarme	86

LISTA DE TABELAS

Tabela 1 – Resumo do regulamento da ANAC	24
Tabela 2 – Resumo das regras por área	26
Tabela 3 – Solicitações de informações e comandos de missões do MAVLink	46
Tabela 4 – Índices do pacote MAVLink	47
Tabela 5 – Resumo das tecnologias utilizadas nos trabalhos relacionados.....	56
Tabela 6 – Coordenadas geográficas da missão	61
Tabela 7 – Custos dos componentes.....	72

LISTA DE ABREVIATURAS E SIGLAS

AFCS	Automated Flight Control System – Sistema Automático de Controle de Voo
AIC	Circular de Informações Aeronáuticas
ANAC	Agência Nacional de Aviação Civil
APM	ArduPilot Mega
BVLOS	Beyond Visual Line of Sight – Além da Linha de Visada Visual
CMA	Certificado Médico Aeronáutico
DC	Direct Current – Corrente Contínua
DECEA	Departamento de Controle do Espaço Aéreo
ESC	Electronic Speed Control – Controle Eletrônico de Velocidade
EVLOS	Extended Visual Line of Sight – Linha de Visada Visual Estendida
FM	Frequency Modulation – Modulação de Frequência
FPV	First Person View – Visão em Primeira Pessoa
GCS	Ground Control Station – Estação de Controle Terrestre
Glonass	Global Navigation Satellite System – Sistema Global de Navegação por Satélite
GPS	Global Positioning System – Sistema de Posicionamento Global
IMU	Inertial Measurement Unit – Unidade de Medição Inercial
MAVLink	Micro Air Vehicle Link – Micro Link para Veículo Aéreo
MP	Mission Planner
PPM	Pulse Position Modulation – Modulação de Posição de Pulso
PWM	Pulse Width Modulation – Modulação de Largura de Pulso
RBAC-E	Regulamento Brasileiro de Aviação Civil Especial
RF	Rádio Frequência
RPA	Remotely Piloted Aircraft – Aeronave Remotamente Pilotada

RPM	Rotações Por Minuto
SARPAS	Solicitação de Acesso de Aeronaves Remotamente Pilotadas
SISANT	Sistema de Aeronaves Não Tripuladas
SITL	Software In The Loop – Software no Laço
VANT	Veículo Aéreo Não Tripulado
VLOS	Visual Line of Sight – Linha de Visada Visual

SUMÁRIO

1 INTRODUÇÃO	16
1.1 Objetivo geral.....	18
1.2 Objetivos específicos.....	19
1.3 Justificativa	19
1.4 Organização do trabalho	20
2 VEÍCULOS AÉREOS NÃO TRIPULADOS.....	22
2.1 Regulamentação de VANTs no Brasil	22
2.2 Regras exclusivas para atividades de recreação	25
2.3 Quadcopters	27
2.4 Graus de liberdade	28
2.5 Modos de voo.....	29
2.6 Hardware.....	29
2.6.1 Sensores	29
2.6.1.1 Giroscópio	30
2.6.1.2 Acelerômetro	31
2.6.1.3 Magnetômetro	31
2.6.1.4 Barômetro	32
2.6.1.5 Sistema de Posicionamento Global (GPS).....	32
2.6.1.6 Unidade de Medição Inercial (IMU).....	33
2.6.2 Controladoras	34
2.6.2.1 ArduPilot Mega (APM).....	34
2.6.2.2 Pixhawk	36
2.6.2.3 Naza	37

2.6.3 <i>Frame</i>	37
2.6.4 Rádio transmissor e receptor.....	39
2.6.5 Motor <i>brushless</i>	40
2.6.6 Controle eletrônico de velocidade (ESC).....	41
2.6.7 Hélice	42
2.6.8 Bateria LiPo	44
2.6.9 Transmissão de imagens	45
2.7 Software.....	45
2.7.1 Telemetria.....	45
2.7.2 ArduPilot / Copter	47
2.7.3 Mission Planner	48
2.7.4 MAVProxy	49
2.7.5 Simulador SITL	49
2.7.6 Python	49
2.7.7 DroneKit.....	50
2.8 Trabalhos relacionados	50
2.8.1 Geração automática de código para a trajetória de VANTs em voos.....	50
2.8.2 Projeto de um sistema de controle para veículos aéreos não tripulados	52
2.8.3 Estudo e desenvolvimento de um <i>quadrotor</i> e sua estação de pilotagem remota	52
2.8.4 Plataforma de desenvolvimento para algoritmos de controle	53
2.8.5 Resumo comparativo dos trabalhos relacionados	55
3 METODOLOGIA.....	57
3.1 Delineamento.....	57
3.2 Cenário	58
3.3 Protótipos	58
3.4 Avaliação da proposta	59
4 DESENVOLVIMENTO.....	60
4.1 Simulações de missões de voo	60
4.2 Desenvolvimento do protótipo do <i>drone</i>	62
4.2.1 Componentes utilizados para a montagem	64
4.2.2 Custos.....	72
4.2.3 Montagem.....	72

4.2.4 Configurações e calibrações	76
4.2.5 Voos de teste	77
4.3 Desenvolvimento do software	78
4.3.1 Envio de missões para o <i>drone</i>.....	79
4.3.2 Desenvolvimento do software de GCS.....	82
4.4 Avaliação da proposta	85
4.5 Resultados	87
4.6 Dificuldades.....	87
5 CONCLUSÃO.....	89
REFERÊNCIAS	91

1 INTRODUÇÃO

Os primeiros Veículos Aéreos Não Tripulados (VANTs) foram construídos para fins militares. Inspirados em bombas voadoras alemãs utilizadas na Segunda Guerra Mundial e nos aeromodelos controlados por rádio, foram projetados para atuar em missões altamente perigosas para seres humanos, em áreas de inteligência militar, assistência e controle de tiro de artilharia, apoio aéreo ao campo de batalha, controle de mísseis de cruzeiro, vigilância urbana, ambiental, costeira e de fronteiras, atividades de busca e resgate, entre outras (CORRÊA, 2008).

Corrêa (2008), cita que o Departamento de Defesa Americano define VANT como um veículo motorizado aéreo que não possui um piloto a bordo. O mesmo utiliza forças aerodinâmicas para sustentar-se no ar, podendo ser controlado por controle remoto ou voar de forma autônoma. Além disso pode ser descartável ou recuperável, tendo capacidade de transportar uma carga útil que pode ser letal. Ainda é definido que veículos balísticos, mísseis de cruzeiro e projéteis de artilharia não são considerados VANTs.

Recentemente, avanços nas tecnologias, relacionados à evolução dos sistemas globais de navegação e geoprocessamento estão expandindo as perspectivas de uso dos VANTs, uma vez que estes equipamentos estão tornando-se relativamente baratos e fáceis de usar (ANDRADE, 2016). Segundo Fowers (2008), a ampliação do uso de VANTs sucede-se pelo fato dos mesmos proporcionarem maior segurança e praticidade quando aplicado às várias situações que anteriormente exigiam uma aeronave de grande porte com um piloto.

Os VANTs de pequeno porte, com pequenos motores e baixa autonomia são conhecidos como *drones*, termo de origem inglesa que é traduzido como zangão ou zumbido, som característico que estes equipamentos emitem através de seus motores e hélices. Devido à aerodinâmica, os *drones* empregam a rotação controlada de seus rotores para garantir sua

estabilidade e condições de voo (SILVA, 2016).

Nos últimos anos, agricultores de diversos países começaram a utilizar *drones* para aplicar os conceitos da agricultura de precisão, com uso de recursos da tecnologia da informação para melhorar a produção agrícola. Sensores e recursos de imagem gradativamente mais precisos e eficientes equipam *drones* que podem auxiliá-los, diminuindo os danos nas lavouras através de levantamentos de dados que possibilitam detectar pragas e calcular o índice de crescimento das plantas. Os *drones* destacam-se entre outros sistemas de monitoramento, pois seus voos são de baixo custo, podendo ser executados com mais frequência, durante todo o período de produção (ANDRADE, 2016).

Andrade (2016), também destaca uma pesquisa realizada no Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (ICMC-USP), onde foi construído um *drone* autônomo para a pulverização de agroquímicos. Baseado em Inteligência Artificial, o sistema é capaz de executar voos de treinamento em diferentes condições meteorológicas, a fim de aprender padrões, para que mais tarde possa tomar decisões no decorrer da pulverização, em condições parecidas com as anteriores. O sistema possui uma estação de controle onde um técnico monitora o andamento dos processos. O protótipo provou ser eficaz, liberando quantidades controladas de agroquímicos, proporcionando uma pulverização com maior precisão e com maior cobertura da aplicação, garantindo dessa forma um aproveitamento melhor dessas substâncias e com menores danos ao ambiente.

Manoel Silva Neto, fundador da Droneng Drones e Engenharia, considera os *drones* como tecnologia consolidada e com aplicações validadas. Ele afirma que os modelos de mercado estão gradativamente mais robustos, com maiores autonomias de voos e com maiores capacidades de cargas, o que proporciona uma expansão ainda maior para a demanda de novas aplicações (SCUSSEL, 2015).

O gerente de vendas da Santiago & Cintra Geo-Tecnologias recomenda o uso de *drones* para mapeamento e inspeção remota, evitando a necessidade de pessoas acessarem locais perigosos, como estruturas altas de uma obra. Documentações fotográficas aéreas, que antes eram feitas com aeronaves tripuladas, agora são feitas com *drones*, com custos de combustível e logística reduzidos (SCUSSEL, 2015).

Atualmente, existem no Brasil várias empresas especializadas no uso de *drones* para topografia e georreferenciamento, aplicações que deram início ao uso dos *drones* na área

tecnológica. Segundo o Engenheiro Wellausen, esse número deve crescer muito pois os conceitos de aerofotogrametria têm-se disseminado e os softwares de modelagem 3D estão mais sofisticados, o que resulta em aplicações mais confiáveis e profissionais (PATELLA, 2016).

Existem vários modelos de *drones* à venda no mercado, como o 3DR Solo, Parrot BeBop e o DJI Phantom. Estes inovaram o mercado, mas aos poucos surgem novos modelos que fornecem voo e controle mais atraentes para iniciantes. Ao mesmo tempo, as comunidades de hardware e software livres também evoluem de forma a desafiar seus rivais mais caros, com recursos avançados, como desvio de obstáculos e navegação pelo Sistema de Posicionamento Global (GPS – *Global Positioning System*). Muitos dos principais fabricantes são bastante ativos e contribuem regularmente nas comunidades de hardwares e softwares abertos (LAIRD, 2016).

Um *drone* pode ser considerado uma plataforma voadora e não está limitado a nenhuma área de atuação específica, uma vez que pode ser usado para várias finalidades, bastando incorporar sensores apropriados (SHARMA, 2016). Segundo Pecharromán e Veiga (2017), em um futuro próximo, à medida que a tecnologia de voos autônomos melhorar, os *drones* poderão realizar tarefas mais complexas, permitindo respostas rápidas e precisas a crises humanitárias e a desafios mundiais, como exemplo mudanças climáticas.

Diante do que foi apresentado e do advento das tecnologias, bem como as demandas de desenvolvimento de equipamentos autônomos para auxiliar em atividades onde existem riscos e dificuldades de acesso pelo ser humano, torna-se imprescindível o aprofundamento no estudo de VANTS do tipo *drone*, mais especificamente na execução de tarefas autônomas, que será o tema do presente trabalho.

1.1 Objetivo geral

O objetivo deste trabalho consiste em construir um VANT do tipo *drone* com a capacidade de realizar voos autônomos, para fornecer meios de verificar e validar incidentes que forem detectados através do disparo de alarmes¹ de dispositivos ou serviços de alerta²,

¹ Comunicação sonora ou visível, direcionada para os responsáveis na iminência de ocorrer um evento perigoso (ZUQUETTE, 2017).

² Avisos que alertam sobre a proximidade de ocorrência de um evento perigoso, para que possam ser tomadas medidas de emergência e de mobilização. É uma condição menos iminente que alarme (ZUQUETTE, 2017).

apresentando dados e imagens para o utilizador, considerando que os referenciais geográficos são previamente conhecidos e que será desenvolvido um software para monitoramento e controle terrestre do *drone*.

1.2 Objetivos específicos

- Realizar um estudo observando as principais tecnologias presentes na maioria dos *drones*, que permitem voos autônomos;
- Definir as tecnologias e recursos necessários para o desenvolvimento do projeto;
- Disponibilizar um referencial teórico e prático para que em futuros trabalhos não seja necessária uma ampla pesquisa para entender o funcionamento do equipamento;
- Desenvolver e configurar o *drone*;
- Desenvolver um software com capacidade de monitorar a posição do *drone*, autorizar a execução de missões e visualizar as imagens transmitidas pelo mesmo.
- Realizar um experimento prático onde serão fornecidos pontos fixos, que representam dispositivos ou serviços de alerta, os quais o *drone* deverá sobrevoar em forma de missão autônoma, caso ocorra o disparo de um alerta ou alarme de incidente.

1.3 Justificativa

O VANT é uma ferramenta muito útil para pesquisadores testarem e avaliarem ideias em várias áreas, como navegação, controle de voo, sistemas em tempo real, Robótica, entre outros. Um VANT do tipo *drone* é bastante versátil como plataforma de testes, tendo como vantagem o custo que é relativamente baixo, a diversidade de tamanhos, mecanismos simples de controle, facilidade em manobras com potencial de pairar e decolar na vertical, sendo possível voar e pousar em pequenas áreas (CHOUHARI; SANKESHWARI, 2016).

O futuro dos *drones*, no uso como ferramentas humanitárias, está rapidamente ganhando forças, principalmente devido ao acesso do setor civil aos mesmos e a variedade de aplicações e soluções possíveis. A área de imagens aéreas, aliada com as tecnologias de voos autônomos, pode agilizar a capacidade de resposta a desastres, uma vez que as imagens ao vivo podem ser

transmitidas para uma equipe remota (PECHARROMÁN; VEIGA, 2017).

Um trabalho realizado por Carvalho e Netto (2014), trata da gestão de áreas de riscos de deslizamentos, realizando um estudo sobre a percepção de risco das populações de locais que utilizam sistemas de alerta e alarme. O artigo caracteriza eventos extremos como fenômenos de dinâmica do relevo terrestre e caracteriza desastres como as consequências destes fenômenos, diante de prejuízos sociais, econômicos e ecológicos. O crescimento urbano desordenado ocasiona diversos problemas socioambientais, infraestrutura deficiente, habitações em áreas de risco e alterações dos sistemas naturais. Na região sudeste do Brasil, estão inseridas as principais áreas de ocorrência de inundações e deslizamentos, sendo estes, exemplos de incidentes ambientais onde o emprego do *drone* pode facilitar o trabalho das equipes de verificação ou resgate.

O presente trabalho busca desenvolver um VANT que seja capaz de auxiliar o trabalho de equipes de verificação de incidentes. Como exemplo, podem ser citados incidentes ambientais, onde o local muitas vezes pode ser de difícil acesso, tornando a verificação muito custosa e dependente de uma equipe disponível no momento do acontecimento. Dessa forma, as equipes serão acionadas somente nos casos onde os incidentes forem confirmados.

O projeto de um VANT é bastante desafiador, pois não envolve somente a área de Computação, mas também de Automação. Segundo Laird (2016), enquanto a disponibilidade de modelos de *drones* foram aumentando no mercado, também aumentaram as comunidades de software e hardware abertos. Este é um dos principais motivos para a decisão de montagem, em vez de compra do *drone*, aliado com a flexibilidade de adquirir somente os componentes necessários, reduzindo assim o custo e o peso do equipamento.

1.4 Organização do trabalho

O presente trabalho está dividido em cinco capítulos, com o propósito de facilitar a leitura e o desenvolvimento dos assuntos envolvidos.

O Capítulo 1: apresenta uma introdução ao assunto, o objetivo geral, os objetivos específicos e as justificativas, elucidando o que deseja-se alcançar com este trabalho.

O Capítulo 2: contempla uma revisão bibliográfica dos principais assuntos abordados neste trabalho, desde as regulamentações de VANTs no Brasil até as teorias fundamentais e os

principais componentes que fazem parte de um *drone*. Também são abordados alguns trabalhos relacionados, com o objetivo principal de conhecer o que já foi desenvolvido por outros autores, na mesma área de estudo.

O Capítulo 3: apresenta a metodologia adotada neste trabalho, as atividades desenvolvidas e também a forma de avaliação dos resultados.

O Capítulo 4: contempla o desenvolvimento do protótipo e da aplicação prática, a avaliação da proposta, os resultados obtidos e as dificuldades encontradas durante o trabalho.

O Capítulo 5: expõe as considerações finais, contemplando as limitações do protótipo desenvolvido e as sugestões para trabalhos futuros.

Por fim, são apresentadas as referências bibliográficas consultadas durante a execução das pesquisas.

2 VEÍCULOS AÉREOS NÃO TRIPULADOS

Neste capítulo são abordados os temas relevantes ao estudo proposto, servindo de embasamento teórico, apresentando os conceitos sobre VANTS, aprofundando em *drones*, abordando os hardwares e softwares necessários para o desenvolvimento do projeto proposto, tendo como objetivo situar o leitor nos conceitos relativos aos mesmos.

2.1 Regulamentação de VANTS no Brasil

Em maio de 2017 a Agência Nacional de Aviação Civil (ANAC) publicou uma regulamentação para a utilização de VANTS, o Regulamento Brasileiro de Aviação Civil Especial (RBAC-E) nº 94, com o objetivo de tornar viável a utilização desses equipamentos, instituindo regras com o objetivo de resguardar a segurança das pessoas, contribuindo para o desenvolvimento seguro e sustentável do setor (BRASIL. RBAC-E 94, 2017).

O regulamento da ANAC categoriza os VANTS em dois tipos:

- Aeromodelos: aeronaves não tripuladas cuja finalidade é de recreação;
- Aeronaves Remotamente Pilotadas (RPAs – Remotely Piloted Aircraft): aeronaves não tripuladas, pilotadas de uma estação remota, com finalidade não recreativa, como por exemplo comercial, corporativa ou experimental.

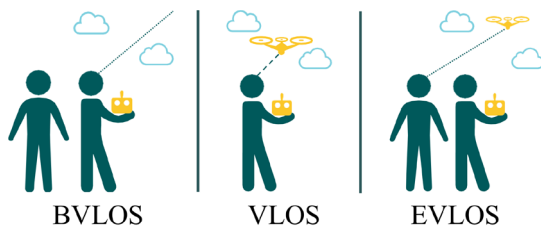
O regulamento deixa claro que aeronaves com mais de 250 g devem voar em um limite de, no mínimo 30 metros distante horizontalmente de pessoas não envolvidas com a operação. Esse limite deve ser seguido quando não existem barreiras de proteção, entre o equipamento e as pessoas, que seja forte o suficiente para protegê-las. Para voar próximo de pessoas é

necessário que estas estejam informadas e que tenham concordado com o voo. Esse limite não é aplicado a operações de órgãos de segurança pública, como polícia, defesa civil, corpo de bombeiros, entre outros.

O regulamento apresenta três modos de operação de voo:

- Operação Além da Linha de Visada Visual (BVLOS – *Beyond Visual Line of Sight*): é quando o piloto não mantém o *drone* no seu campo de visão, nem mesmo o observador³;
- Operação em Linha de Visada Visual (VLOS – *Visual Line of Sight*): é quando o piloto é capaz de visualizar o *drone* a olho nu;
- Operação em Linha de Visada Visual Estendida (EVLOS – *Extended Visual Line of Sight*): é quando o piloto remoto só é capaz de visualizar o *drone* com a ajuda de lentes ou de outros equipamentos, sendo necessária a ajuda de um observador.

Figura 1 – Modos de operações de voo



Fonte: Adaptado pelo autor com base em Regras da ANAC (2017, p. 4).

O regulamento ainda classifica as RPAs baseado nos pesos máximos de decolagem:

- Classe 1: equipamentos que possuem mais que 150 kg. Precisam ser certificados através de um processo semelhante ao de certificação de aeronaves tripuladas e necessitam de cadastro no Registro Aeronáutico Brasileiro, sendo identificados com sua marca e matrícula. Os voos devem ser registrados;
- Classe 2: possuem entre 25 kg e 150 kg. Também necessitam de cadastro no Registro Aeronáutico Brasileiro, sendo identificados com sua marca e matrícula. Todos os voos devem ser registrados;

³ Pessoa que auxilia o piloto remoto, sem operar o equipamento (BRASIL. RBAC-E 94, 2017).

- Classe 3: engloba os equipamentos entre 250 g e 25 kg. Equipamentos dessa classe que operem no modo BVLOS ou acima de 120 m são obrigados a realizar o cadastro no Registro Aeronáutico Brasileiro (identificados com sua marca e matrícula) e ainda deverão pertencer a um projeto autorizado pela ANAC. Se operarem até 120 m e em VLOS somente precisam de registro na ANAC, pelo Sistema de Aeronaves Não Tripuladas (SISANT). Os voos não precisam ser registrados.

Não existe idade mínima para pilotar aeromodelos, porém a idade mínima exigida para pilotos e observadores de RPAs é de 18 anos. Além disso, são exigidas também licença e habilitação (emitidas pela ANAC) para pilotos de RPAs de classes 1, 2 ou 3 que desejam voar acima do limite de 120 m. Pilotos das classes 1 e 2 devem possuir também o Certificado Médico Aeronáutico (CMA)⁴. A Tabela 1 expõe um resumo do regulamento da ANAC.

Tabela 1 – Resumo do regulamento da ANAC

	RPAs Classe 1	RPAs Classe 2	RPAs Classe 3	Aeromodelos
Registro da aeronave?	Sim	Sim	BVLOS: Sim VLOS: Sim	Sim
Aprovação ou autorização do projeto?	Sim	Sim	Apenas BVLOS ou acima de 400 pés	Não
Limite de idade para operação?	Sim	Sim	Sim	Não
Certificado médico?	Sim	Sim	Não	Não
Licença e habilitação?	Sim	Sim	Apenas para operações acima de 400 pés	Apenas para operações acima de 400 pés

Fonte: Adaptado pelo autor com base em Regras da ANAC (2017, p. 7).

É proibido o transporte de artigos perigosos que não possuem relação com atividades agrícolas, florestais e que não sejam requeridos para a operação do equipamento. Em voos comerciais, equipamentos eletrônicos que possuem baterias de lítio somente podem ser transportados se são utilizados durante o voo, como exemplo, câmeras de foto e vídeo, computadores, entre outros.

⁴ É um exame de saúde pericial, cujo objetivo é verificar a aptidão física e mental de tripulantes de aeronaves ou pilotos remotos de VANTS. Fonte: ANAC (2018). Disponível em <<http://www.anac.gov.br/assuntos/setor-regulado/profissionais-da-aviacao-civil/processo-de-licencas-e-habilitacoes/a-cma-2013-exame-medico>>. Acesso em: 29 abr. 2018.

2.2 Regras exclusivas para atividades de recreação

No dia 11 de junho de 2018, começaram a vigorar as regras da Circular de Informações Aeronáuticas (AIC) N 17/18, publicada pelo Departamento de Controle do Espaço Aéreo (DECEA) e intitulada “Aeronaves Remotamente Pilotadas para uso Recreativos Aeromodelos”. Este regulamento objetiva instruir os usuários que desejem voar, com o propósito de recreação, no Espaço Aéreo Brasileiro com suas RPAs (BRASIL. AIC-N 17/18, 2018).

A AIC N 17/18, estabelece o princípio da sombra que é o volume existente ao redor de qualquer estrutura ou obstáculo, artificial ou natural, sendo 5 m verticalmente acima e trinta metros horizontalmente para os lados da estrutura. Este volume não faz parte do espaço aéreo, pois não é possível ser utilizado por aeronaves tripuladas. Por tanto, é permitido o uso deste volume para a prática de aeromodelismo, desde que sejam seguidas as regras da circular.

As áreas ou locais adequados para os aeromodelos utilizarem o espaço aéreo brasileiro devem ser afastados de áreas muito populosas e de instalações vulneráveis a barulhos, como hospitais, casas de repouso, escolas e templos religiosos, além de cumprir também os seguintes parâmetros (BRASIL. AIC-N 17/18, 2018):

- Devem localizar-se distantes de zonas de aproximação e de decolagem de aeródromos, sendo 45° para cada lado do eixo da pista, a um afastamento mínimo de 9 km a partir da cabeceira da pista mais próxima;
- Devem localizar-se, no mínimo, a 2 km distantes de aeródromos cadastrados, medidos da parte mais próxima da área patrimonial do aeródromo. Caso exista algum obstáculo mais próximo do que os 2 km, poderão ser realizados voos de aeromodelos do lado oposto ao do circuito de tráfego dos aeródromos em uma distância de até 30 m do obstáculo, não ultrapassando a altura do mesmo;
- Os voos não podem ser realizados a um raio de menos de 600 m da coordenada central de helipontos⁵ cadastrados.

Em zonas urbanas os voos em VLOS ou BVLOS só são permitidos quando utilizando

⁵ Área homologada e demarcada oficialmente para o pouso e decolagem de helicópteros (BRASIL. AIC-N 17/18, 2018).

Visão em Primeira Pessoa (FPV – *First Person View*)⁶ dentro do princípio da sombra. Devem ser respeitados os limites de altitude, de acordo com as áreas pretendidas, com a finalidade de não afetar as rotas conhecidas de aeronaves e helicópteros tripulados e corredores visuais. Voar a uma altura máxima de 40 m e uma velocidade máxima de 40 km/h. Distanciamento horizontal mínimo de 30 m de construções, veículos, animais de terceiros e pessoas não anuentes.

Em zonas rurais é preciso manter um afastamento mínimo de 2 km de áreas onde são realizadas operações de aviação agrícola. Operações em VLOS e BVLOS somente são permitidas quando utilizando o FPV. O limite de altura é de 50 m acima do obstáculo sobrevoado e o limite máximo de velocidade é de 100 km/h. É preciso manter, no mínimo, 90 m de afastamento de pessoas não anuentes, construções, veículos, aeronaves em solo e animais de terceiros (BRASIL. AIC-N 17/18, 2018).

Áreas adequadas são os espaços propícios para a prática de aeromodelismo, que possuem as distâncias requeridas de aeródromos e rotas de aeronaves tripuladas. Operações em VLOS e BVLOS somente são permitidas quando utilizando o FPV. A altura máxima é de 120 m, caso seja necessário ultrapassar esse limite, como por exemplo para eventos de competição, será necessário solicitar permissão para o órgão regional subordinado ao DECEA. A Tabela 2 consiste em um resumo das principais regras do DECEA com relação aos tipos de áreas.

Tabela 2 – Resumo das regras por área

Áreas	Altura	Limite de velocidade	Distância de pessoas não anuentes	FPV autorizado
Zonas urbanas	40 m ¹	40 km/h	30 m	Não ²
Zonas rurais	50 m	100 km/h	90 m	Sim
Áreas adequadas	≤120 m ³	Não aplicável ⁴	Não aplicável ⁵	Sim

Fonte: Adaptado pelo autor com base em Brasil. AIC-N 17/18 (2018). ¹ No princípio da sombra a altura máxima passa a ser definida pelo obstáculo. ² O FPV pode ser utilizado dentro do princípio da sombra. ³ Altura máxima é estabelecida por meio de uma análise. ⁴ É necessária uma atenção aos limites horizontais das áreas. ⁵ Entende-se que as pessoas dentro destas áreas são anuentes com as operações ali realizadas.

As regras para registro da aeronave na ANAC continuam valendo conforme previsto no RBAC-E nº 94, o qual determina que todos os aeromodelos cujos pesos máximos de decolagem ultrapassem 250 g, operados somente em VLOS, com altura máxima de 120 m e que não sejam

⁶ É uma forma de controlar o equipamento remoto que permite ao piloto ver da perspectiva da aeronave, que possui uma câmera e um transmissor de vídeo. O streaming do vídeo ao vivo é transmitido para um óculos de vídeo ou uma tela portátil. Fonte: PÉREZ-SÁNCHEZ, Héctor A. et al. *Cockpit Design for First Person View Flight for a Remotely Operated Quadrotor Helicopter*. *Computación y Sistemas*, [s. l.], v. 19, n. 3, 2015. Disponível em: <<http://dx.doi.org/10.13053/CyS-19-3-2014>>. Acesso em: 29 abr. 2018.

de projetos autorizados, devem ser registrados no SISANT. Além disso, será necessário o cadastro no sistema de Solicitação de Acesso de Aeronaves Remotamente Pilotadas (SARPAS), onde devem ser realizados os cadastros de voos recreativos que pretendem ser realizados em locais externos as áreas adequadas para esta prática.

A AIC-N 17/18 alerta os pilotos a não voarem em condições meteorológicas adversas, com pouca visibilidade e alturas inadequadas, além de não transportar artigos considerados perigosos, que podem constituir riscos à saúde, segurança e ao meio ambiente. Os voos devem ser imediatamente cancelados em casos que forem constatadas aproximações de aeronaves tripuladas.

2.3 Quadcopters

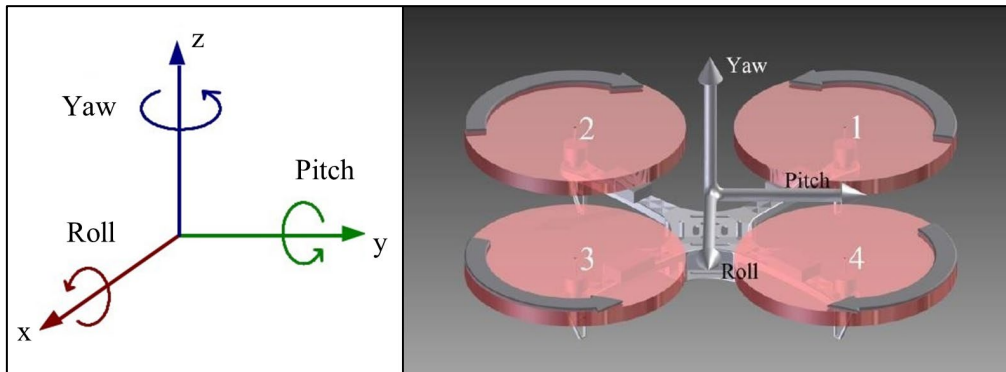
Um *quadcopter*, também conhecido popularmente por *drone*, é um tipo de VANT, cujo movimento é controlado a partir da aceleração ou desaceleração de seus conjuntos, motor e hélice. Essencialmente, um *drone* é um robô voador que pode ser controlado remotamente ou voar de forma autônoma através de planos de voo controlados por uma Estação de Controle Terrestre (GCS – *Ground Control Station*). Os VANTs têm sido frequentemente associados aos militares, mas também são usados para busca e resgate, vigilância, monitoramento de tráfego, monitoramento do clima e combate de incêndios, entre outras coisas (BREGU, 2015).

Segundo Melo (2010), no aeromodelismo o movimento vertical ocorre através do comando de aceleração (*throttle* em inglês), subindo ou descendo. O movimento na horizontal pode ocorrer através de arfagem (*pitch* em inglês) que é quando a aeronave é inclinada para a frente ou para trás, ou pode ocorrer através de rolagem (*roll* em inglês) que é quando a aeronave é inclinada para um lado ou outro. Por último e não menos importante, o movimento de guinada (*yaw* em inglês) que é quando a aeronave gira em torno do eixo vertical.

Segundo Choudhari e Sankeshwari (2016), a estrutura de um *quadcopter* é simples, sendo composta basicamente por quatro rotores, ou seja, dois pares de conjunto motor e hélice, sendo que cada par gira em um sentido, horário e anti-horário. Essas rotações combinadas geram um torque, que faz com que as hélices produzam uma força vertical de elevação para cima, elevando o *quadcopter* no ar ou fazendo mover-se. Os movimentos de inclinação e rotação ocorrem alterando a velocidade de um dos pares de motores, enquanto o outro par permanece em velocidade constante. O movimento de guinada pode ser alcançado alterando a

velocidade de ambos os pares de motores. A Figura 2 ilustra os eixos de rotação e a estrutura de um *drone quadcopter*, o posicionamento dos motores e os eixos de rotação.

Figura 2 – Eixos de rotação



Fonte: Carrilho (2016, p. 34)⁷.

2.4 Graus de liberdade

É possível constatar que um objeto está em movimento se, em diferentes instantes, a sua posição for diferente, caso contrário encontra-se em repouso. Para descobrir se a posição é diferente é necessário usar outros objetos como referência para medir a sua posição. O movimento é relativo, uma vez que o objeto pode estar movimentando-se em relação a um objeto de referência e em repouso em relação a outro objeto (VILLATE, 2013).

Os graus de liberdade de um objeto, ou de um sistema, podem ser definidos como as variáveis utilizadas para medir a sua posição. Por exemplo, a fim de definir o posicionamento de uma mosca em uma sala, pode-se medir a distância da mesma até duas paredes perpendiculares na sala e medir a distância até o chão, resultando em um sistema de três coordenadas cartesianas, podendo ser representadas pelas letras x, y e z. Porém, essa mosca ainda pode mudar sua orientação, podendo girar em cada um dos três eixos, resultando assim em seis graus de liberdade. Além disso, uma mosca pode abrir ou fechar asas e pode esticar ou dobrar seu corpo, então possui ainda mais graus de liberdade (VILLATE, 2013).

De forma geral, os *quadcopters* possuem seis graus de liberdade, três são os deslocamentos lineares ou translação nos eixos x, y e z. Os outros são os movimentos de rotação

⁷ CARRILHO, Bruno Caceres. **Plataforma Web para Caracterização de Drones**. 2016. 94 f. TCC (Graduação) - Curso de Engenharia Elétrica Com ênfase em Eletrônica, Escola de Engenharia de São Carlos da Universidade de São Paulo, São Carlos, 2016. Disponível em: <<http://www.tcc.sc.usp.br/tce/disponiveis/18/180450/tce-12012017-152608>>. Acesso em: 02 jun. 2018.

em torno dos mesmos eixos, *pitch* (rotação no eixo y), *roll* (rotação no eixo x) e *yaw* (rotação no eixo z), apresentados na Seção 2.3 (DEMOLINARI, 2016).

2.5 Modos de voo

Segundo Bregu (2015), o modo de voo é uma função que gerencia os dados enviados para os motores, por exemplo, no modo *land* (terra em inglês), a aeronave desacelera os motores até pousar. Existem modos de voo que utilizam, e outros que não utilizam, recursos do GPS. Entre os modos de voos mais comuns podem ser citados:

- *Stabilize* (estabilização): permite ao piloto comandar a aeronave manualmente, porém com controle automático de nivelamento;
- *Altitude hold* (manter altitude): possui as mesmas características do *stabilize* e ainda mantém a altitude atual. Utiliza um sensor barométrico ou sonar para obter a altura;
- *Loiter*: utiliza o GPS para manter automaticamente a posição, direção e altitude atual. O piloto pode voar como se estivesse no modo manual e ao liberar as alavancas no rádio transmissor a aeronave manterá a posição;
- *Auto*: a aeronave executará missões pré-programadas. Este modo utiliza o controle de altitude do modo *altitude hold* e o controle de posição do modo *loiter*;
- *Return to launch* (retornar ao lançamento): utiliza o GPS para fazer com que a aeronave retorne para a sua posição de partida.

2.6 Hardware

O hardware de um *drone* é composto pela parte elétrica, que contém motores, baterias e barramentos. E também é composto pela parte eletrônica, que contém sensores, placa controladora, *Electronic Speed Control* (ESC) e rádio receptor (MELO, 2010). Esses são os principais componentes de um *drone* e são apresentados nessa seção.

2.6.1 Sensores

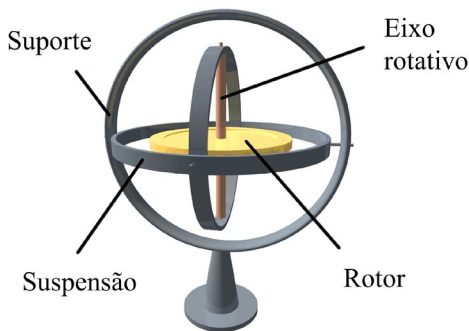
Devido ao advento dos semicondutores e da tecnologia da informação, os processadores e sensores estão ficando cada vez mais baratos, com mais funcionalidades e maior precisão.

Isso reduziu o custo de sistemas embarcados, dando origem a vários robôs e VANTs (SUN, 2012). Segundo Andrade (2012), um sensor é um dispositivo eletrônico cuja função é medir valores como temperatura, altura, distância, entre outros. Esta seção apresenta os sensores mais comuns utilizados em *drones* ou em suas placas controladoras, que analisam os dados obtidos pelos sensores para que o desempenho nos voos seja o melhor possível.

2.6.1.1 Giroscópio

Inventado em 1852 por Léon Foucault, o giroscópio é um dispositivo que baseia-se em princípios físicos newtonianos, sendo bastante empregado na navegação marítima e na aviação. O giroscópio clássico, do tipo mecânico, apresentando na Figura 3, consiste em um rotor que gira em torno de seu próprio eixo, suspenso por um suporte que contém dois círculos articulados (CLARK apud OLIVEIRA; SILVA; BARBOSA, 2008).

Figura 3 - Giroscópio mecânico



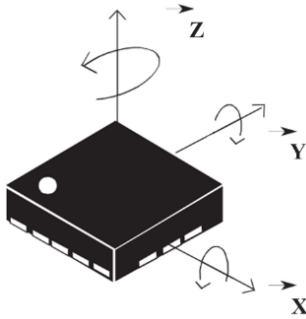
Fonte: Adaptado pelo autor com base em Oliveira, Silva e Barbosa (2018, p.1).

A atividade do giroscópio baseia-se no princípio da inércia, onde o eixo em rotação mantém a direção fixa em relação ao espaço. Enquanto os círculos giram, a inércia da rotação mantém o dispositivo em sua posição de início e a ação desta força tende a impedi-lo de mudar de direção durante um rápido movimento, devido à resistência proporcionada pela inércia (RAND apud OLIVEIRA; SILVA; BARBOSA, 2008).

De acordo com o mesmo autor, o giroscópio mecânico tornou-se inviável em aplicações reais, devido ao atrito mecânico causado pelo movimento de rotação do mesmo, que diminuía progressivamente a ação da inércia no decorrer da atuação. O dispositivo, então foi deixado de lado até o final do século XIX, quando com o advento dos motores elétricos, ressurgiu na forma de um dispositivo do tipo sensor que atualmente é imprescindível na navegação marítima, aérea e em voos espaciais. A Figura 4 ilustra um giroscópio digital com seus referentes eixos de

rotação.

Figura 4 – Giroscópio digital



Fonte: Adaptado pelo autor com base em Santos (2015, p. 21).

2.6.1.2 Acelerômetro

O acelerômetro é um dispositivo eletromecânico desenvolvido para realizar a medida da aceleração de um determinado corpo, quando sofre uma força externa. Essa força pode resultar em acelerações dinâmicas, por exemplo a variação de velocidade de um deslocamento ou vibrações e choques. Também pode resultar em aceleração estática, por exemplo, quando o dispositivo é inclinado e a força de aceleração gravitacional terrestre é projetada sobre os eixos de sensibilidade. O acelerômetro converte um vetor de aceleração em um sinal elétrico, podendo ser coletado e processado por sistemas eletrônicos (SILVA, 2013).

Segundo Barato (2014), o acelerômetro capacitivo é formado por duas microestruturas, ambas muito próximas, existindo assim uma capacitância entre as mesmas. Caso se aplique uma força no dispositivo, esta capacitância sofre alteração. Este tipo de acelerômetro possui aplicações em diversas áreas, como exemplo em *smartphones*, identificação de terremotos, monitoramento de vibrações de motores e monitoramento de comportamento de animais.

2.6.1.3 Magnetômetro

O magnetômetro é um dispositivo empregado na medição do campo magnético da Terra. É utilizado para medir a direção, o sentido e a intensidade de campos magnéticos. Basicamente dividem-se em dois tipos: escalar e vetorial. O magnetômetro do tipo escalar mede a magnitude do campo magnético, enquanto o vetorial mede a magnitude e o sentido do campo magnético na direção em que está alinhado. Utilizando dois magnetômetros vetoriais, não paralelos entre si, mas paralelos à Terra, é possível realizar o cálculo da direção do norte magnético do planeta, a partir da soma vetorial das leituras (GABRIEL, 2014).

2.6.1.4 Barômetro

É um dispositivo utilizado para verificar pressões atmosféricas. Na meteorologia é utilizado para prever mudanças no clima conforme a variação da pressão. Na área da Robótica, bem como nos *drones*, o dispositivo pode ser utilizado como altímetro para medir a altura do mesmo, uma vez que a pressão varia de acordo com a altitude (SANTOS, 2015).

2.6.1.5 Sistema de Posicionamento Global (GPS)

O Sistema de Posicionamento Global (GPS), surgiu através da rivalidade armamentista entre a União Soviética e os Estados Unidos, com o intuito de obter a posição exata e em tempo real de algum objeto, seja pessoa, veículo, míssil, navio, entre outros. Os soviéticos desenvolveram o Sistema Global de Navegação por Satélite (Glonass – *Global Navigation Satellite System*), e em 1957 lançaram o satélite Sputnik I, dando início a corrida espacial (ROCHA, 2003).

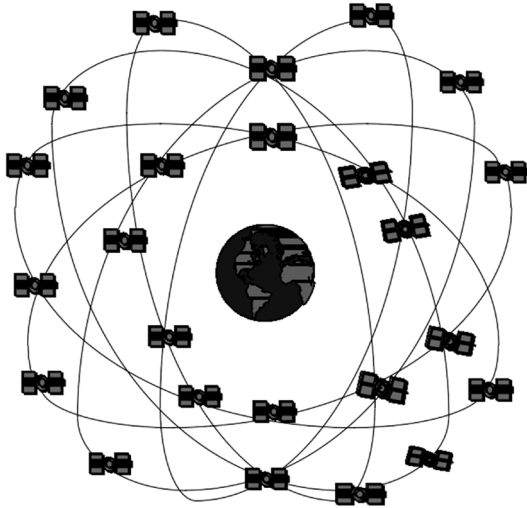
As ondas de rádio transmitidas pelo Sputnik I acabaram sendo observadas pelos cientistas norte-americanos, que constataram que através do efeito Doppler era possível determinar a órbita do satélite e com a posição do satélite conhecida, era possível calcular a posição do receptor das ondas de rádio transmitidas pelo satélite. O Departamento de Defesa Americano deu início ao projeto do GPS em 1973, surgindo como uma expansão ao já existente sistema de navegação da Marinha Americana. Mais tarde, em 1978 foi lançado o primeiro satélite de GPS (ROCHA, 2003). Segundo Monico (2008), em 1985 o GPS foi declarado operacional, contando com 24 satélites em órbita. No final de 2005 haviam 29 satélites operacionais e em junho de 2007 haviam trinta satélites em operação.

O GPS permite que qualquer usuário possua a seu dispor, no mínimo quatro satélites para serem rastreados, seja qual for a sua localidade da superfície terrestre. A navegação por GPS baseia-se na medida de distâncias entre o usuário e estes quatro satélites. Uma vez que as posições dos satélites são conhecidas, são calculáveis as coordenadas da antena do usuário. Geometricamente, apenas três distâncias seriam suficientes para calcular as coordenadas, porém uma quarta medida é indispensável, em razão da não sincronia entre os relógios dos satélites e o relógio do utilizador (MONICO, 2008).

Em uma altitude de aproximadamente 20.200 km existem, no mínimo 24 satélites, dispostos em seis planos orbitais (representados na FIGURA 5), sendo que cada plano possui

quatro satélites. Estes planos orbitais encontram-se inclinados 55° relativos ao Equador, com período orbital que aproxima-se de 12 horas siderais. Desta forma, a posição de cada satélite repete-se, a cada dia, com uma antecipação de aproximadamente quatro minutos em relação ao dia anterior. Por este motivo, sempre existem, no mínimo quatro satélites GPS disponíveis para qualquer local da superfície da terra, a qualquer instante (MONICO, 2008).

Figura 5 – Constelação dos satélites GPS



Fonte: Adaptado pelo autor com base em Monico (2008, p. 40).

Em resumo, o GPS é um sistema de navegação por rádio baseado em satélites que permite que objetos em terra, mar e ar designem suas posições tridimensionais (latitude, longitude e altitude), além de sua velocidade e tempo (ROCHA, 2003). Tratando de VANTs, Andrade (2012), afirma que o GPS é importante na realização de missões autônomas, devendo ser usado em espaços abertos para possibilitar a captura de sua localização.

2.6.1.6 Unidade de Medição Inercial (IMU)

Uma Unidade de Medição Inercial (IMU – *Inertial Measurement Unit*), é um circuito eletrônico que contém acelerômetros, giroscópios e muitas vezes um magnetômetro, que proporcionam informações sobre o posicionamento do VANT para a controladora de voo. A partir da política de controle e ação, o Sistema Controlador de Voo Automático (AFCS – *Automated Flight Control System*), utiliza as informações da IMU para calcular as respostas dos motores. A maior parte das controladoras de voo mais modernas já possuem IMUs incorporadas as suas arquiteturas (PIXHAWK apud DEMOLINARI, 2016).

2.6.2 Controladoras

A prática de voo com *drones* pode ser complicada, sendo inviável a sua operação em modo completamente manual, devido a quantidade de variáveis envolvidas, como motores com diferenças, hélices desbalanceadas, turbulências, falhas elétricas, vibrações mecânicas e até falhas elétricas e mecânicas. Os microcontroladores têm chegado ao mercado gradativamente mais eficientes, rápidos e com preços mais acessíveis, permitindo o desenvolvimento de AFCS ou piloto-automático (SUN apud DEMOLINARI, 2016).

O piloto-automático de um VANT tem o objetivo de manter o voo estabilizado, seja durante o voo, a decolagem ou a aterrissagem. Esse controle é feito através de cálculos da dinâmica, respondendo a quaisquer mudanças percebidas através da leitura de dados dos sensores (ARTALE apud DEMOLINARI, 2016). Segundo o mesmo autor, esse sistema de percepção e respostas é chamado de política de controle e está relacionado aos modos de voos (Seção 2.5). Muitas controladoras de voo modernas possuem parâmetros ajustáveis pelo usuário, permitindo personalizar os modos de voo da aeronave, por exemplo, em aplicações de filmagens aéreas pode ser configurado um voo mais estável e suave, ao mesmo tempo que para *drones* de competição pode ser ajustada uma política com respostas mais rápidas, que traga como consequências um voo mais instável e até menos seguro.

Em resumo, a controladora é responsável por capturar os sinais dos sensores (acelerômetro, giroscópio, barômetro, etc.), capturar os canais do rádio receptor (*throttle*, *pitch*, *roll* e *yaw*), executar algoritmos de controle de estabilização de voo, controlar os motores, comunicar com GCS, entre outras (MELO, 2010). Segundo Sun (2012), a principal função do AFCS é permitir que a operação da aeronave seja econômica e segura, que as missões de voos possam ser realizadas e que eventos inesperados possam ser manipulados. Esta seção aborda algumas das controladoras existentes no mercado, começando com a ArduPilot Mega (APM).

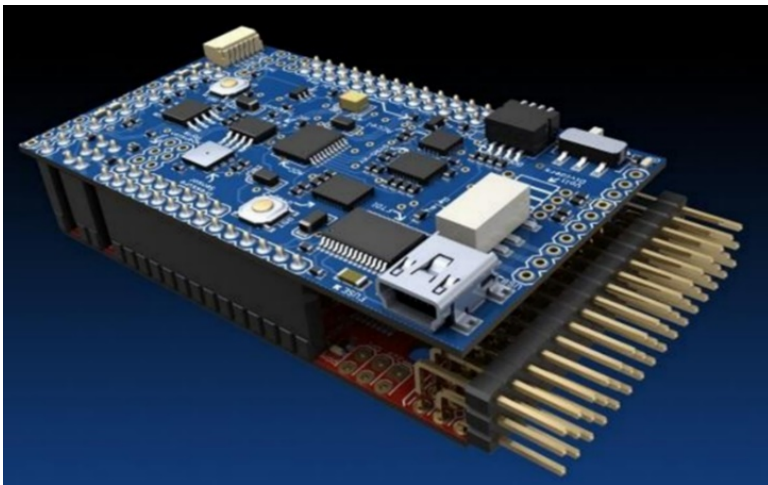
2.6.2.1 ArduPilot Mega (APM)

A ArduPilot Mega (APM) faz parte do sistema ArduPilot (apresentado na Seção 2.7.2) e foi desenvolvida com o intuito de ser aplicada no desenvolvimento de mecanismos de piloto automático para VANTs. Sendo um dos sistemas de piloto automático de código-fonte aberto pioneiro na utilização de seis graus de liberdade (GABRIEL, 2014). Segundo Andrade (2012), existem duas versões da APM, a versão 1 que é a mais antiga e é composta por duas partes e a

versão 2, que é mais recente, possui mais sensores e um tamanho menor.

A placa vermelha da APM 1 contém o processador Atmega 2560 e um coprocessador Atmega328, sendo esta, responsável pelo controle e pelo processamento dos sinais recebidos do rádio controle através de 8 canais de entrada PWM. Nesta placa é encaixada a placa azul que inclui vários sensores como giroscópio, acelerômetro, sensores de pressão e tensão, funcionando como um IMU que possibilita a estabilidade do *drone* (ANDRADE, 2012).

Figura 6 – APM 1



Fonte: Barato (2014, p. 22).

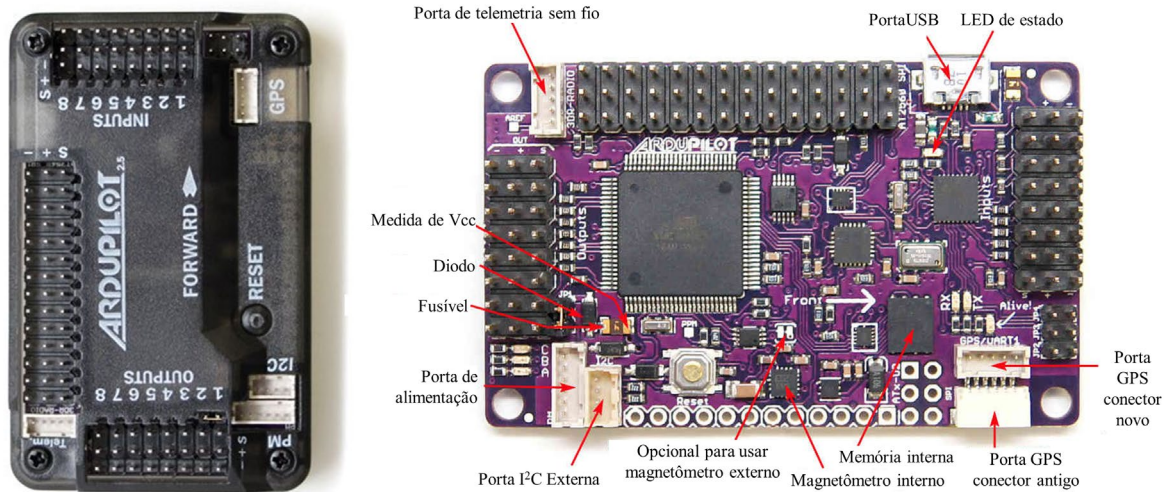
Na versão 2.5 da APM, os componentes eletrônicos e os sensores estão todos na mesma placa. Barato (2014), cita as principais características:

- Microcontrolador ATmega2560 com *clock* de 16 MHz;
- Circuito de segurança contra falhas, utilizando um circuito com um chip multiplexador e um processador ATmega32U2;
- Oito canais de entrada do rádio, com opção de ligar e desligar o piloto automático;
- Conectores para adicionar sensor de GPS e outros sensores;
- Fabricado em uma placa de quatro camadas;
- Giroscópio para os três eixos;
- Barômetro para altitude.

A APM 2.6 é a última versão da placa e não é mais suportada pelas novas versões do

*firmware*⁸ Copter (3.3 e posteriores), descrito na Seção 2.7.2. A Figura 7 ilustra a APM2.6.

Figura 7 – APM 2.6



Fonte: Adaptado pelo autor com base em Ardupilot (2016)⁹.

2.6.2.2 Pixhawk

A Pixhawk é uma controladora que surgiu do projeto Pixhawk (MEIER et al., 2012), que baseou-se em desenvolver um *quadcopter* cujo principal objetivo era fornecer uma plataforma de pesquisa para controle e voo autônomo, usando visão computacional. O equipamento deveria ser poderoso o suficiente para lidar com todos os processos de imagem e controle de voo a bordo.

A controladora Pixhawk foi então desenvolvida pela 3DRobotics em parceria com o grupo Computer Vision and Geometry Lab. Seu projeto é de código-fonte aberto e possui suporte de uma comunidade voluntária de desenvolvedores. Está equipada com sensores de giroscópio, acelerômetro, magnetômetro e barômetro, todos de três eixos. Possui um processador Cortex M4F de 168 MHz e 32 bits, que é superior ao da APM. Além de plataformas de *drones*, ela ainda suporta aviões, carros, robôs bípedes e barcos (ALMEIDA, 2015).

⁸ Software gravado na memória de programa de um microcontrolador ou memória *flash*, evitando que toda a vez que o dispositivo for ligado o mesmo tenha que ser recarregado (MELO, 2010).

⁹ Disponível em <<http://ardupilot.org/copter/docs/common-26-overview.html#common-26-overview>>. Acesso em 28 mar. 2018.

Figura 8 – Pixhawk



Fonte: Almeida (2015, p. 32).

2.6.2.3 Naza

A Naza é uma controladora produzida pela empresa DJI que, diferente da APM e da Pixhawk, possui seu código-fonte e eletrônica fechada. Sendo assim, não é possível de forma legal ao usuário realizar modificações na mesma. A Naza possui avançados algoritmos de estabilização e um modo de *failsafe*¹⁰, porém só suporta *drones* e não suporta tantos softwares e plataformas como as controladoras citadas anteriormente (ALMEIDA, 2015).

Figura 9 – Naza-M Lite e módulo GPS/Bússola



Fonte: DJI apud Almeida (2015, p. 33).

2.6.3 Frame

A estrutura do *drone*, também chamada de *frame* (quadro em inglês), é constituída por uma parte central, onde geralmente são fixados os sensores, as baterias e a controladora de voo.

¹⁰ É um mecanismo de segurança utilizado em situações onde o *drone* perde a conexão com o controle remoto. Permite, por exemplo, que a aeronave retorne a sua posição de lançamento ou continue a missão, dependendo de como for configurado (ANDRADE, 2012).

Nesta parte, também são encaixados os braços do *frame*, cujas extremidades recebem o acoplamento dos motores. Segundo Demolinari (2016), as placas que formam a parte central do *frame* devem ser rígidas, suportando os esforços e mantendo o formato da estrutura em voo. Já Choudhari e Sankeshwari (2016), afirmam que toda a estrutura deve ser a mais rígida possível, a fim de evitar um voo instável.

Existem *frames* com configurações de três, quatro, seis e oito hélices, chamados respectivamente de *tricopters*, *quadcopters*, *hexacopters* e *octocopters*. Dessa forma, existem alguns detalhes determinantes na escolha do *frame*, em especial no que refere-se ao seu peso e ao peso extra de materiais que precisam ser transportados. Quanto mais hélices o *drone* conter, mais peso ele consegue transportar, além de uma maior estabilidade no ar. Um *drone* do tipo *quadcopter* apresenta uma ótima agilidade no ar, ao ser comparado com um *drone* do tipo *hexacopter*, que executa manobras lentas (SANTOS, 2015). A Figura 10 demonstra um *frame* de *drone quadcopter*.

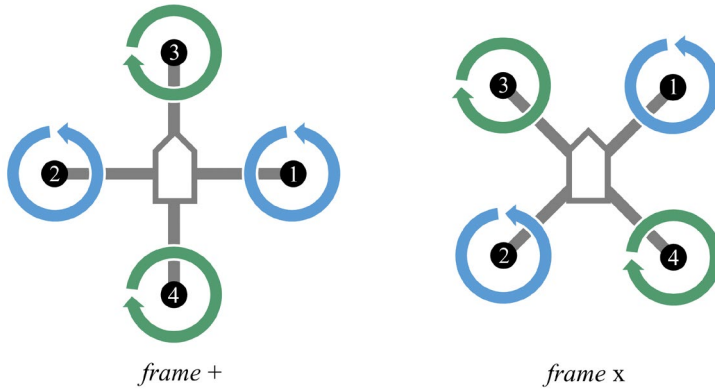
Figura 10 – *Frame* modelo DJI F450



Fonte: Santos (2015, p. 10).

Um *quadcopter* pode ter um *frame* configurado em forma de “+” ou de “x”. O *frame* em formato de “x” geralmente é utilizado quando existe uma câmera agregada ao *drone*, evitando que a mesma colida com o braço inclinado para a frente durante um movimento de *pitch* (ANDRADE, 2012). A Figura 11 representa as duas configurações presentes em *drones* do tipo *quadcopter* e os sentidos de rotações das hélices.

Figura 11 – Configuração de *frame* em “+” ou em “x”

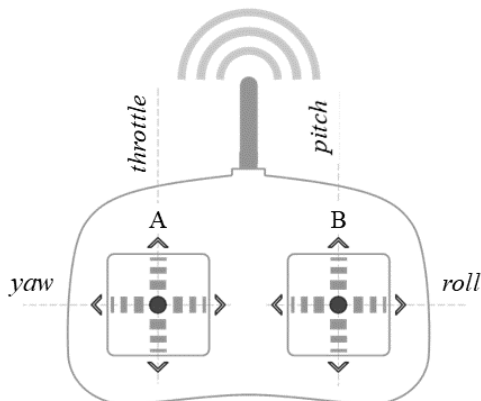


Fonte: Adaptado pelo autor com base em Andrade (2012, p. 21).

2.6.4 Rádio transmissor e receptor

No aeromodelismo, o controle de uma aeronave dá-se através de um link de Rádio Frequência (RF) do tipo Modulação de Frequência (FM – *Frequency Modulation*) constituído por um rádio transmissor e um rádio receptor, cujas frequências podem ser de 72 MHz ou 2.4 GHz, que são liberadas para este fim pelo órgão competente, no caso do Brasil, a ANATEL. Os comandos mais básicos são dados através de duas alavancas (*sticks* em inglês), uma controlada pela mão esquerda e outra pela mão direita. Cada *stick* pode ser empurrado para frente ou para trás e para à direita ou para à esquerda (MELO, 2010). A Figura 12 ilustra um rádio transmissor.

Figura 12 – Rádio transmissor



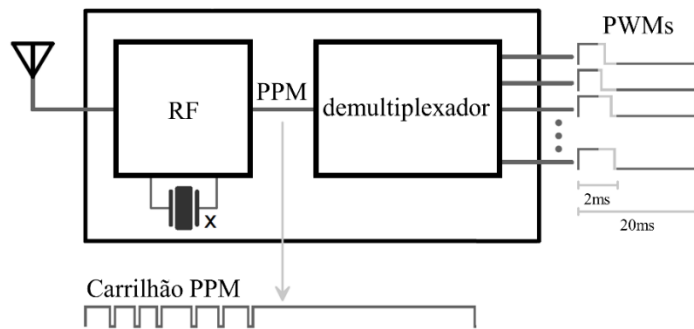
Fonte: Adaptado pelo autor com base em Melo (2010, p. 27).

Melo (2010), explica que os comandos de voo são modulados por Modulação de Posição de Pulso (PPM – *Pulse Position Modulation*), no qual a posição de cada *stick* no transmissor é representada por um pulso invertido com largura de 0,3 ms (pode variar conforme o transmissor) com término assumindo qualquer momento entre 1 ms e 2 ms. Os comandos são enviados em canais, a cada 20 ms, serialmente em formato de carrilhão (canal 1, canal 2, e

assim sucessivamente). Após o envio dos canais, o sinal mantém-se em nível lógico alto e, 0,3 ms anteriormente ao fim do quadro de 20 ms, o sinal volta para o nível lógico baixo, com a intenção de sincronismo e início de um novo carrilhão.

O rádio receptor (ilustrado na FIGURA 13), é dividido em dois blocos: o de RF e o de demultiplexação. O bloco de RF remove a portadora usada no sinal recebido com o objetivo de recompor o carrilhão de canais. O outro bloco executa a demultiplexação do carrilhão de canais em sinais de Modulação de Largura de Pulso (PWM – *Pulse Width Modulation*), que possui um período de 20 ms, devido ao intervalo de envio da sequência pelo rádio transmissor. O intervalo de tempo mínimo que um sinal de PWM mantém-se em nível lógico alto (normalmente 5 V) é de 1 ms e o máximo é de 2 ms. No resto do tempo, o sinal preserva-se em nível lógico baixo até completar os 20 ms. Sinais PWM podem ser utilizados como sinais de entrada em controles eletrônicos de velocidade, para controlar os motores *brushless*, assunto da Seção 2.6.5.

Figura 13 – Rádio receptor



Fonte: Adaptado pelo autor com base em Melo (2010, p. 28).

2.6.5 Motor *brushless*

Segundo Homa (2013), motores são máquinas que geram energia mecânica através de outros tipos de energia, sendo que o motor elétrico transforma energia elétrica em energia mecânica. Complementa ainda que em aeronaves à hélices, o motor não gera tração diretamente, mas através das hélices.

Os motores de Corrente Contínua (DC – *Direct Current*) sem escova (*brushless*) são semelhantes aos motores com escova (*brushed*), porém não existe um contato mecânico para a passagem de eletricidade entre o estator (enrolamento fixo) e o rotor (ímã na parte externa que gira em torno do seu próprio eixo). Esse tipo de motor oferece muitas vantagens em relação aos motores com escovas e aos motores à indução, principalmente no que diz respeito a eficiência

e funcionamento suave. Os motores sem escova ganharam notoriedade nas indústrias automotivas, eletrodomésticas, aeroespaciais, entre outras. Algumas vantagens deste tipo de motor (FOUR et al., 2007):

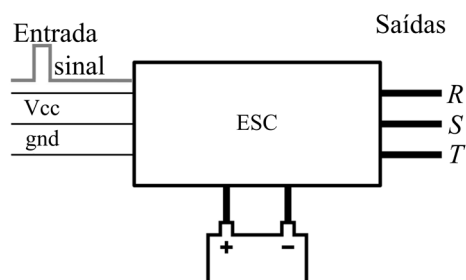
- Alta potência, baixa inércia, alta taxa de torque e alta resposta dinâmica;
- Alta eficiência;
- Vida operacional longa e alta confiabilidade;
- Sem geração de pó proveniente do desgaste de escovas;
- Baixo ruído;
- Operação em altas velocidades;
- Baixa resistência térmica.

Segundo Melo (2010), os motores do tipo *brushless* são alimentados através de um circuito de controle eletrônico de velocidade, apresentado na Seção 2.6.6.

2.6.6 Controle eletrônico de velocidade (ESC)

O Controle Eletrônico de Velocidade (ESC – *Electronic Speed Control*) é um circuito eletrônico que, além de fornecer energia aos enrolamentos do motor a partir de uma fonte DC, também controla a velocidade do mesmo (MELO, 2010). Segundo Santos (2015), são frequentemente utilizados em aeronaves rádio controladas movidas a eletricidade. A Figura 14 ilustra um ESC, de forma generalista.

Figura 14 – ESC genérico



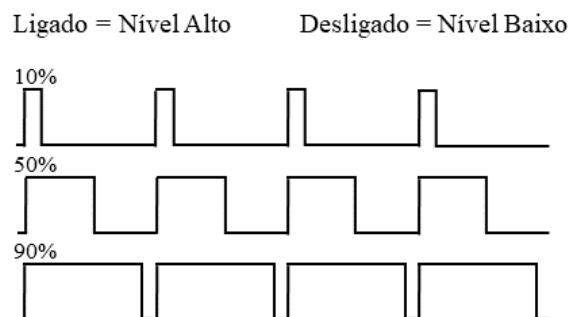
Fonte: Adaptado pelo autor com base em Melo (2010, p. 31). As saídas R, S e T são os terminais a serem conectados às três fases do motor *brushless*.

Basicamente, um ESC pode ser separado em uma parte de controle e outra parte de potência. A primeira parte é formada por um microcontrolador, que através de um sinal PWM (gerado no rádio receptor) com período de 20 ms, gera três outros, normalmente trapezoidais com períodos variando conforme o fabricante, mas tipicamente 8 kHz e 16 kHz. A segunda parte, cujas entradas são os sinais gerados pela primeira, amplifica a corrente das três ondas (fases) por meio de transistores de efeito de campo (dois por enrolamento), que mesmo possuindo capacidade de corrente considerável para fornecer energia aos enrolamentos do motor *brushless*, também atuam como chaves que ligam ou desligam o enrolamento, baseado no sinal de PWM produzido pelo microcontrolador (MELO, 2010).

De acordo com Santos (2015), a velocidade do motor, medida em Rotações Por Minuto (RPM), é exatamente proporcional à tensão aplicada em seus terminais, sendo o torque diretamente proporcional ao fluxo de corrente através do mesmo. A tensão pode ser controlada por meio do uso de sinais PWM para recortar a corrente do motor, proporcionando a aceleração necessária. O processo é realizado através do chaveamento da alimentação, ligando e desligando em uma certa frequência, variando o tempo de permanência, tempo conhecido como ciclo de trabalho (*duty cycle* em inglês).

A Figura 15 ilustra a forma de onda de 10 %, 50 % e 90 % de *duty cycle*, deixando claro que um *duty cycle* de 10 % significa estar em nível alto em 10 % do comprimento da onda e em nível baixo em 90 % do tempo dela. Como resultado do PWM, a tensão enviada para o motor pode ser definida a partir de desligado (0 % de *duty cycle*) até ligado (100 % de *duty cycle*), com uma boa eficiência (FLORIANO apud SANTOS, 2015).

Figura 15 – Diagrama de PWM



Fonte: Adaptado pelo autor com base em Floriano apud Santos (2015, p. 17).

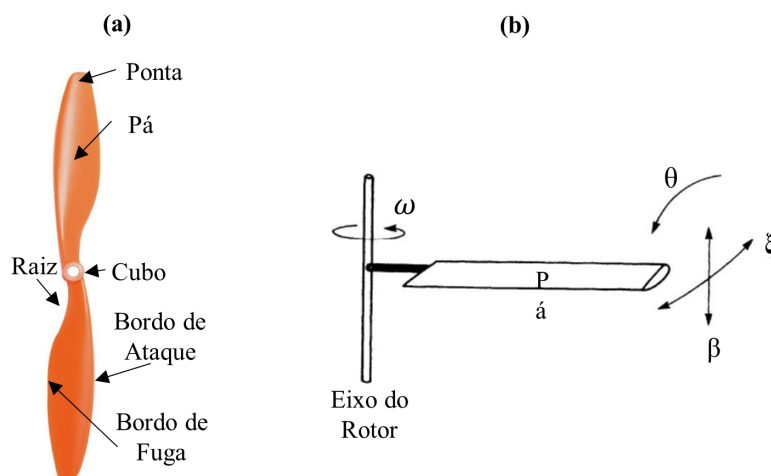
2.6.7 Hélice

A hélice é uma peça do grupo motopropulsor que fornece a tração, transformando a

potência real do motor em potência útil. Uma hélice é constituída de duas ou mais pás, tendo um perfil aerodinâmico semelhante ao da asa de um avião (HOMA, 2013).

Em um *drone*, as hélices operam sob severas condições, agindo sobre elas o torque dos motores e a força de arrasto aerodinâmico da propulsão do ar. Elas também sofrem rápidas variações de aceleração para manter a estabilidade do equipamento, por isso, elas necessitam de alta resistência, associada a baixo custo, uma vez que são quebradas facilmente em acidentes (JOHNSON apud DEMOLINARI, 2016). A Figura 16 ilustra as características de uma hélice.

Figura 16 – Características de uma hélice



Fonte: Adaptado pelo autor com base em Johnson apud Demolinari (2016, p. 30). (a) Partes principais da hélice e (b) ângulos de interesse.

As hélices de *drones* normalmente possuem um passo fixo, o que os difere de helicópteros comerciais e até de alguns aeromodelos. O passo está relacionado ao ângulo de seção (β) da hélice, que corresponde ao quanto a hélice está inclinada em relação a seu eixo longitudinal central naquela seção transversal. O passo representa o deslocamento dessa hélice quando realizado um giro completo, porém não considerando o arrasto aerodinâmico e a compressão do ar. Por exemplo, hélices de helicópteros de passageiros possuem um ângulo de seção constante, pois as pás de suas hélices são planas. Hélices de *drones* normalmente possuem um ângulo de seção variável por toda a extensão da pá (BARROS apud DEMOLINARI, 2016).

Para *drones* do tipo *quadcopter*, são recomendadas hélices de passo pequeno, a fim de obter melhor estabilidade e menores vibrações. Os tamanhos sugeridos das hélices geralmente são encontrados nas especificações dos motores, caso contrário devem ser calculadas, ou mesmo testados diferentes tamanhos de hélices. Hélices maiores resultam em maior eficiência de empuxo do que hélices menores, podendo transportar mais carga útil, porém também será preciso mais energia para aumentar o tempo de voo (YADAV; SHARMA; BORAD, 2017).

O mesmo autor define que a eficiência de uma hélice pode ser medida em relação ao avanço (J), que é a relação entre a velocidade de avanço da aeronave (V) e a rotação da hélice (n) em rotações por segundo. A equação matemática para calcular a eficiência da hélice ainda leva em consideração o diâmetro da hélice (D), conforme a equação 1.

$$J = \frac{V}{nD} \quad (1)$$

2.6.8 Bateria LiPo

Devido à grande popularidade das baterias de íons de lítio (Li-Ion) no final dos anos 90, foi possível desenvolver um tipo de bateria que utiliza os mesmos íons de lítio, entretanto divididos por uma camada de polímero, com poros micrométricos, umedecida com um solvente orgânico que servia de eletrólito. A partir da modernização dos processos de fabricação, foi possível desenvolver filmes microscópicos de lítio, eletrólito (polímero e solvente) e o anodo de um material compósito. Estes filmes formam novas células de baterias que atualmente são conhecidas como baterias de LiPo, acrônimo de lítio e polímero (VINCENT; SCROSATI apud DEMOLINARI, 2016).

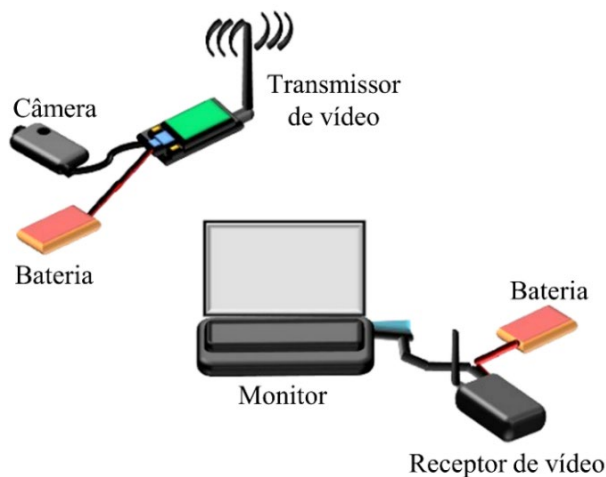
Entre as principais propriedades de uma bateria LiPo, podem ser citadas: a carga, a taxa de descarga, o número de células e o peso. A carga representa a quantidade de energia armazenada nas células eletroquímicas, cuja unidade de medida é o ampère-hora (Ah), que equivale a 3600 Coulomb. A taxa de descarga, representada pela letra C, indica a quantidade de corrente que a bateria pode fornecer, sendo que uma bateria de 1000 mAh e 2 C tem capacidade de fornecer uma corrente máxima de 2000 mA (2 x 1000). O número de células ligadas em série, representado pela letra S, especifica quantos grupos de baterias menores existem no conjunto, sendo que cada grupo é definido como uma bateria LiPo. Cada célula deve ser descarregada até um limite de 80 % da capacidade total, ou seja, cerca de 3.7 V, uma vez que cada célula chega a 4.2 V quando totalmente carregada (DEMOLINARI, 2016).

Em comparação com outras fontes de energia do tipo bateria, as do tipo LiPo fornecem grande densidade de energia e possui uma alta taxa de descarga, tornando-as ideais para uso em *drones* (SHARMA, 2016). Segundo Demolinari (2016), os *drones* operam em altas correntes, por isso existe uma grande exigência por baterias com estas características.

2.6.9 Transmissão de imagens

A captura de imagens em *drones* é realizada através de uma câmera a bordo. Essas imagens chegam até o transmissor de vídeo (geralmente um rádio FM de 5.8 GHz, 2.4 GHz, 900 MHz, entre outros), onde são moduladas e convertidas em sinais analógicos modulados, que são enviados pelo ar, sendo propagado até um receptor de vídeo. O receptor de vídeo demodula os sinais analógicos em dados analógicos capturados (também realiza alguns filtros de ruídos), que podem ser exibidos em um monitor, óculos de FPV ou outro equipamento que exibe imagens (UBAYA; IQBAL, 2014). A Figura 17 ilustra os componentes envolvidos na captura e transmissão de vídeo.

Figura 17 – Hardware de captura e transmissão de vídeo



Fonte: Adaptado pelo autor com base em Ubaya e Iqbal (2014, p. 42).

2.7 Software

Na composição e funcionamento de um VANT são envolvidos diversos softwares, desde o *firmware* gravado na memória de programa do equipamento e a GCS que monitora os voos até os protocolos de comunicação. Estes softwares constituem o assunto desta seção.

2.7.1 Telemetria

Segundo Neto (2016), a telemetria é responsável pela comunicação serial entre o sistema de navegação da aeronave e a GCS. A mesma pode ser dada através de um enlace de rádio de 915 MHz utilizando o protocolo de comunicação Micro Link para Veículo Aéreo (MAVLink – *Miniature Air Vehicle Link*), que possui código-fonte aberto e está sendo utilizado por vários softwares de GCSs gratuitas como DronePlanner, GroundControl e Mission Planner (MP).

O MAVLink é um protocolo leve que utiliza mensagens compostas somente por cabeçalhos e permite a comunicação de até 256 VANTs na mesma faixa de frequência. Lançado em 2009 sob a licença GNU GPL, permite controlar VANTs de alto nível, em vez de interagir com elementos de baixo nível, como orientação e empuxo. As mensagens MAVLink podem ser divididas em duas categorias: solicitações de informações e comandos de missão. O protocolo possui uma biblioteca de solicitações de informação e comandos de missão, da qual as principais (representadas na Tabela 3), são consideradas suficientes para controlar o VANT, operando de forma autônoma sob a orientação de um software de GCS (FULLER, 2014).

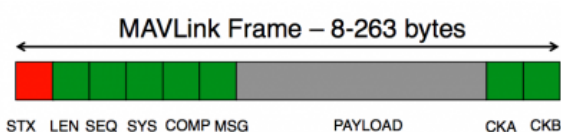
Tabela 3 – Solicitações de informações e comandos de missões do MAVLink

Solicitação de informação	Comando de missão
Ângulo <i>pitch</i>	Navegação para posição GPS
Ângulo <i>yaw</i>	<i>Loiter</i> em um local por tempo indefinido
Ângulo <i>roll</i>	<i>Loiter</i> em um local por tempo definido
Latitude GPS	Retornar ao local de lançamento
Longitude GPS	Decolar
Altitude do solo	Definir modo do sistema
Altitude do nível do mar	Alterar local da casa
Velocidade da altitude	Calibrar sensores
Velocidade da longitude	Desligar componente
Graus da bússola	
Tempo desde a inicialização	

Fonte: Adaptado pelo autor com base em Fuller (2014, p. 3).

O sistema de controle usa o módulo gerador de cabeçalho para traduzir as informações da mensagem para o formato específico do MAVLink. O mesmo gerencia as funcionalidades específicas do MAVLink, como formatação de mensagens, comprimentos variáveis de mensagens, geração de soma de verificação e definição de dispositivo de destino. Quando a geração é concluída, a mensagem formatada é retornada no formato texto para o sistema de controle (FULLER, 2014). A Figura 18 representa um pacote MAVLink e a Tabela 4 descreve cada índice do pacote.

Figura 18 – Pacote MAVLink



Fonte: Fuller (2014, p. 4).

Tabela 4 – Índices do pacote MAVLink

Índice	Conteúdo	Valor	Explicação
0	Sinal de início de pacote	v1.0: 0xFE (v0.9: 0x55)	Indica o início de um novo pacote
1	Comprimento de carga útil	0 - 255	Indica o comprimento de carga útil a seguir
2	Sequência de pacotes	0 - 255	Cada componente conta sua sequência de envio. Permite detectar a perda de pacotes.
3	ID do sistema	1 - 255	ID do sistema de envio. Permite diferenciar outros equipamentos na mesma rede.
4	ID do componente	0 - 255	ID do componente de envio. Permite diferenciar componentes do mesmo sistema, por exemplo, a IMU e o piloto automático.
5	ID da mensagem	0 - 255	ID da mensagem - o id define o que a carga útil “significa” e como ela deve ser decodificada corretamente.
6 para (n+6)	Dados	0 - 255	Os dados da mensagem dependem do ID da mensagem.
(n + 7) para (n+8)	Checksum (byte baixo, byte alto)		Inclui MAVLINK_CRC_EXTRA (calculado a partir dos campos da mensagem). Protege o pacote da decodificação de uma versão diferente do mesmo pacote, mas com variáveis diferentes.

Fonte: QGroundControle (2018). Disponível em: <<http://qgroundcontrol.org/mavlink/start>>. Acesso em 28 mar. 2018.

2.7.2 ArduPilot / Copter

O ArduPilot é um sistema de piloto automático desenvolvido por um grupo de engenheiros e cientistas da Computação. O software é capaz de controlar diversos tipos de veículos, como aviões, *drones*, helicópteros, barcos e até submarinos. Possui ferramentas sofisticadas de registro de dados, análise, simulação e, segundo Ardupilot (2016)¹¹, já foi instalado em mais de 1.000.000 de veículos em todo o mundo. O código-fonte do ArduPilot é aberto, o que contribui para sua rápida evolução. O pacote do software é instalado em aeronaves de várias empresas, como 3DR, jDrones, PrecisionHawk, AgEagle e Kespry. Também é utilizado para testes e desenvolvimento por várias instituições e corporações, como NASA, Intel e Insitu/Boeing, além de universidades em todo o mundo (ARDUPILOT, 2016).

Copter é um *firmware* de código-fonte aberto que faz parte do ArduPilot e é utilizado

¹¹ Disponível em: <<http://ardupilot.org/about>>. Acesso em: 10 abr. 2018.

em *drones*, helicópteros e outros veículos de rotor. Fornece vários modos de voos, desde modos totalmente manuais até modos totalmente autônomos. Funciona com uma variedade de softwares de GCSs que são usados para configurar o veículo, monitorar o voo do veículo em tempo real e realizar atividades de planejamento de missões. O Copter também inclui outras partes do ecossistema ArduPilot, como simuladores, ferramentas de análise de log e APIs de alto nível para controle de veículos (ARDUPILOT, 2016)¹².

2.7.3 Mission Planner

Uma GCS é uma Estação de Controle Terrestre capaz de controlar e monitorar um VANT em tempo real. O software Mission Planner (MP) é uma plataforma que fornece um software para GCS com interface gráfica capaz de configurar o VANT e exibir o estado dos sensores, além de criar e monitorar missões autônomas. Durante os voos, o Mission Planner comunica-se com a aeronave através do protocolo de comunicação MAVLink que dá-se através de um sistema de telemetria (ANDRADE, 2012). A Figura 19 demonstra a tela principal do MP.

Figura 19 – Tela principal do Mission Planner



Fonte: Do autor (2018). À esquerda é exibido um horizonte artificial e abaixo os parâmetros do voo. À direita, é exibido um mapa com a posição do VANT.

Segundo Andrade (2012), o mapa apresenta a posição atual do VANT se o sensor de GPS estiver ativo e sincronizado, ou através do uso de simulador. O horizonte artificial indica a inclinação do *drone* e logo abaixo são exibidos parâmetros do voo, como altitude, velocidade, distância da coordenada para a qual está movendo-se, entre outros.

¹² Disponível em: <<http://ardupilot.org/copter/docs/introduction.html>>. Acesso em 10 abr. 2018.

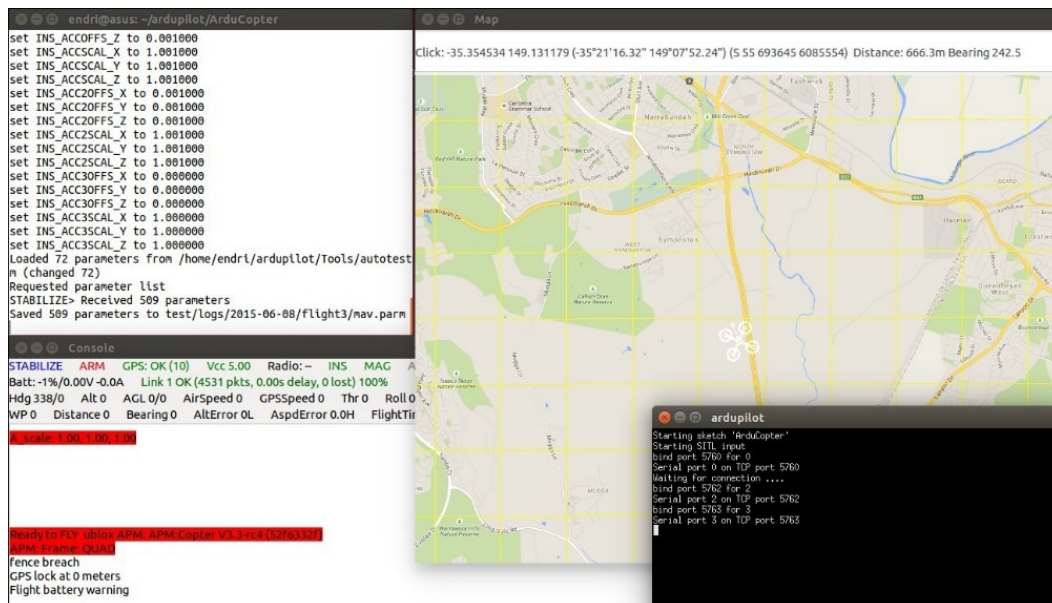
2.7.4 MAVProxy

O MAVProxy é um software GCS de código aberto que roda em linha de comando. Baseia-se na linguagem de programação *Python* e permite a comunicação e pilotagem de VANTs que suportam o protocolo MAVLink. Ele permite que quase todas as funções do Mission Planner sejam executadas através de comandos inseridos no terminal (MARTY, 2014).

2.7.5 Simulador SITL

O simulador Software no Laço (SITL - *Software In The Loop*) permite simular a plataforma ArduPilot sem precisar adquirir um hardware. O simulador trata-se de uma compilação do ArduPilot que utiliza um compilador C++ para emular o hardware da placa APM ou Pixhawk no nível de registro, o que permite simular sensores como giroscópios, acelerômetros e GPS da mesma maneira como realmente funcionam. A Figura 20 ilustra o simulador executando, sendo exibido um *prompt* de comando do MAVProxy no qual é passado instruções para a aeronave simulada, apresentando um mapa que exibe a localização da mesma e um console que exibe informações do voo em tempo real (BREGU, 2015).

Figura 20 – Simulador SITL



Fonte: Bregu (2015, p. 23).

2.7.6 Python

De propósito geral, o Python é uma linguagem de programação que, segundo Lutz e Ascher (2007), combina suporte para Programação Orientada a Objetos e um modo de

desenvolvimento rápido e flexível que faz com que seja constantemente aplicado em funções de *script*, arquivo de nível superior e mais simples do que programa, que é um aplicativo de vários arquivos mais sofisticados. A linguagem assume no mercado uma grande base de usuários e uma comunidade de desenvolvedores muito ativa. As aplicações mais comuns de utilização do Python são programação de sistemas, interfaces gráficas com usuários, *scripts* de Internet, integração de componentes, programação de banco de dados, composição rápida de protótipos e outros. O Python tem código-fonte aberto e é incluído automaticamente em distribuições Linux (LUTZ; ASCHER, 2007).

2.7.7 DroneKit

A DroneKit é uma API de código aberto desenvolvida em Python que permite o desenvolvimento de aplicativos para computadores que comunicam-se com veículos através do protocolo MAVLink. A API fornece acesso as informações de estado e parâmetros, permitindo o gerenciamento de missões e controle direto sobre o movimento e operações do veículo. A DroneKit tem a finalidade de melhorar o piloto automático, dando maior inteligência ao comportamento do veículo, uma vez que tarefas computacionalmente mais complexas podem estar sendo executadas fora do veículo (3D Robotics, 2016)¹³.

2.8 Trabalhos relacionados

Nesta seção são apresentados os trabalhos relacionados ao tema deste trabalho, sendo que alguns deles também ajudaram a fundamentar a pesquisa. Estes trabalhos enfatizam a importância do estudo nesta área que, segundo a maioria dos autores, vem crescendo rapidamente e ganhando diversos tipos de aplicações.

2.8.1 Geração automática de código para a trajetória de VANTs em voos

Almeida (2015), relata em seu estudo que a maioria das controladoras de VANTs possuem *firmwares* que estão preparados somente para utilizar sensores de acelerômetro, giroscópio e GPS. Esse fato limita os VANTs na execução de missões pré-definidas, pois a orientação utilizando GPS só é possível em locais abertos, os quais possuem várias adversidades, seja por obstáculos físicos ou ofuscamento do sinal de GPS. A partir destas

¹³ Disponível em <<http://python.dronekit.io/about/overview.html>>. Acesso em 12 out. 2018.

constatações o autor propôs a criação de um modo de voo automático, onde a missão principal é particionada em mini missões, as quais possuem em alguns pontos da rota (pré-definidos pelo usuário) tomadas de decisão a partir da leitura dos sensores.

Almeida (2015) criou novos módulos em um software chamado Nadzoru, um para particionar uma missão principal em mini missões e outro para a importação e exportação de missões do Mission Planner. O autor desenvolveu também um modo automático no MP que possui a característica de escolher a mini missão a ser enviada para o VANT, com base na leitura de sensores (o autor utilizou um sensor virtual).

Quando é ativado o modo automático criado por Almeida (2015), o Mission Planner busca pela mini missão inicial e a envia para o VANT, que entra em modo de voo automático e executa a missão até chegar ao ponto de decisão, então entra em modo *loiter*, mantendo automaticamente a posição, direção e altitude. Então o MP verifica se existem mini missões, caso exista o sensor é lido e o MP decide qual é a próxima mini missão e a envia para o VANT, que novamente entra em voo automático para executá-la, caso não exista uma mini missão, a missão chegou ao seu fim.

O autor considerou que o projeto teve um desempenho satisfatório, sendo que o VANT realizou missões tomando decisões impostas pelo usuário somente nos pontos pré-selecionados, validando o trabalho através de testes de simulador e em voo real. A Figura 21 apresenta o *drone* utilizado por Almeida (2015) para realizar os testes fora do simulador, montado em um *frame* do tipo *hexacopter* com placa controladora APM 2.6.

Figura 21 – *Drone* hexacopter



Fonte: Almeida (2015, p.67).

2.8.2 Projeto de um sistema de controle para veículos aéreos não tripulados

Barato (2014), observou que os VANTs estão popularizando-se tão rápido quanto a quantidade de funcionalidades exercidas por eles. Paralelamente a isso, o sistema de controle embarcado necessita cada vez mais de melhorias, aumentando a eficiência e a segurança das missões para as quais são designados. Com base nisso, Barato (2014), propôs o aperfeiçoamento de um hardware já existente, a placa APM, utilizada por diversos pesquisadores e aeromodelistas, com o objetivo de substituir o uso da mesma, podendo servir também para instigar futuros trabalhos nesta mesma área.

O projeto substituiu vários sensores utilizados na APM, como giroscópio, bússola, acelerômetro e barômetro, por versões mais atuais. Além disso, acrescentou um sensor de pressão diferencial, utilizado para a medição da velocidade do *drone* em relação ao ar e um GPS integrado (na APM é externo). Segundo o autor, essas modificações reduzem os custos e as dimensões do sistema, uma vez que integra em apenas uma placa todos os sensores necessários.

Barato (2014), concluiu que obteve sucesso no projeto de um novo hardware para ser utilizado no controle de VANTs, composto por sensores mais modernos que a placa já existente, na qual baseou-se, além de adicionar novos sensores já integrados. O autor ressalta que a placa também pode ser empregada no controle de veículos terrestres.

2.8.3 Estudo e desenvolvimento de um *quadrotor* e sua estação de pilotagem remota

Diante da infinidade de aplicações possíveis para VANTs, Chimenes (2017), observou ser necessário o desenvolvimento de métodos que facilitem o uso dos mesmos, então propôs a construção de um *drone quadcopter* e uma estação de pilotagem remota. O autor iniciou realizando um estudo das aeronaves remotamente pilotadas, desde suas aplicações até o funcionamento da mesma.

A estação de pilotagem remota baseou-se em um software de computador responsável por ler uma porta serial conectada a um Arduino que é encarregado pelo enlace de telemetria. As informações são obtidas por outra placa Arduino a bordo da aeronave que realiza a leitura dos dados da controladora de voo, no caso a Naza-M Lite. A comunicação se dá através de um módulo de RF de 2.4 GHz e as imagens são transmitidas pela própria câmera a bordo do *drone*, que cria uma rede sem fio para a transferência das imagens. A Figura 22 ilustra o *drone* do tipo

quadcopter montado pelo autor.

Figura 22 – *Drone* e seu rádio transmissor



Fonte: Chimenes (2017, p. 59).

Como resultados, o autor ressaltou ter construído um *drone* muito robusto, com uma considerável sustentação de carga extra e uma boa autonomia de voo. Concluiu também que a estação de pilotagem auxilia o piloto com informações importantes para sua tomada de decisões durante a execução dos voos.

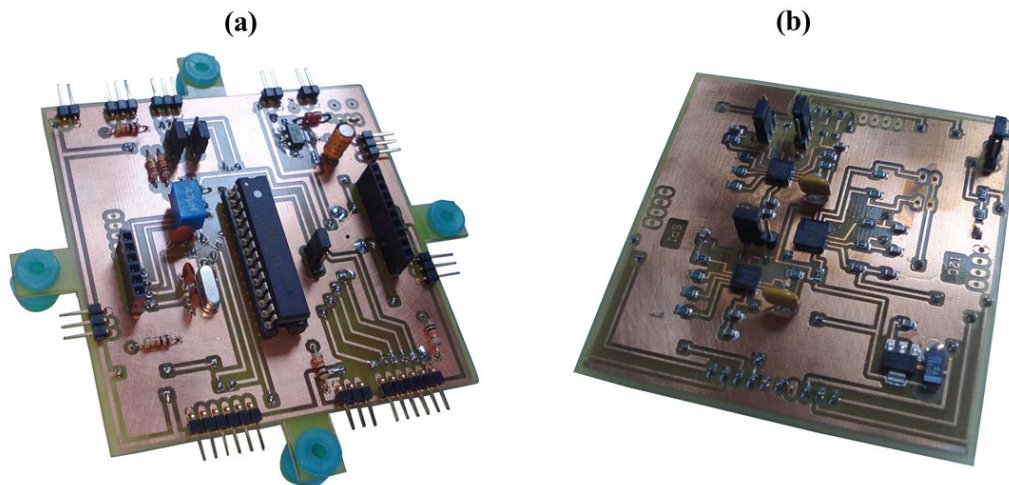
2.8.4 Plataforma de desenvolvimento para algoritmos de controle

Melo (2010), propôs em seu trabalho a implementação de um pequeno VANT, de baixo custo e que tenha autonomia de tomar decisões baseadas em dados dos sensores a bordo e no poder de computação do mesmo. O objetivo do VANT é de ser utilizado como plataforma de desenvolvimento, permitindo a construção e testes de técnicas de controle de estabilização de voo, podendo compor uma pesquisa mais abrangente, como por exemplo o uso de algum tipo de inteligência artificial que pode ser treinada por ações de controle realizadas por um humano.

Diferente do trabalho apresentado na Seção 2.8.3, onde foi utilizada uma placa controladora comercial, Melo (2010) projetou e construiu uma placa controladora que foi dividida em duas placas (FIGURA 23), uma que contempla o microcontrolador e outra os sensores (destacando os acelerômetros e os giroscópios). O autor adquiriu os motores *brushless*, os ESCs e os rádios (transmissor e receptor), pois estes possuem preços bastante acessíveis e são compatíveis com o projeto proposto. O restante da eletrônica foi construída a partir de

componentes elementares, como resistores, capacitores, circuitos integrados, entre outros.

Figura 23 – Placas desenvolvidas por Melo



Fonte: Adaptado pelo autor com base em Melo (2010, p. 41, p.45). (a) Placa com o microcontrolador e (b) placa de sensores.

Melo (2010) atingiu os objetivos do seu trabalho pois desenvolveu com sucesso as placas e o *firmware*, necessários para o desenvolvimento de uma controladora de estabilização de voo. Além disso, o autor observou que algumas melhorias devem ser feitas no futuro, como a escolha de um microcontrolador com mais memória de programa e a troca dos ESCs por outros que disponibilizem entradas digitais usando barramento I²C¹⁴, que possibilitará uma placa mais compacta. A Figura 24 ilustra o VANT desenvolvido por Melo (2010).

Figura 24 – Drone desenvolvido por Melo



Fonte: Melo (2010, p. 18).

¹⁴ I²C (*Inter-Integrated Circuit*): padrão de comunicação serial de duas linhas, onde a informação trafega por uma linha bidirecional, normalmente chamada de SDA (*Serial Data*), a uma taxa de transmissão binária imposta por um sinal de *clock* na outra linha, normalmente chamada de SCL (*Serial Clock*) onde cada dispositivo presente neste barramento, mestre ou escravo, tem um endereço na faixa de 0 a 127 (MELO, 2010).

2.8.5 Resumo comparativo dos trabalhos relacionados

Cada um dos trabalhos analisados apresentou uma abordagem bem particular, enriquecendo o conteúdo desta pesquisa.

O primeiro trabalho, intitulado “Geração automática de código para a trajetória de VANTs em voos”, desenvolvido por Almeida (2015), mostra que mesmo em tecnologias consolidadas, como é o caso do GPS, também existem falhas e situações adversas, mas que estas falhas podem ser contornadas, no caso do GPS, através do uso de outros sensores que podem servir para auxiliar na tomada de decisões.

No segundo trabalho, de título “Projeto de um sistema de controle para veículos aéreos não tripulados”, Barato (2014) sentiu a necessidade de aperfeiçoamento dos hardwares existentes no mercado e diferente do primeiro trabalho, ele não criou uma nova placa do zero, mas a projetou baseando-se na melhoria da placa APM, por ser um projeto aberto e já utilizado por diversos pesquisadores. O autor mostrou que sempre é possível melhorar algo e que projetos abertos permitem que isso aconteça, sem ter que reinventar algo totalmente novo.

O terceiro trabalho, intitulado “Estudo e desenvolvimento de um *quadrotor* e sua estação de pilotagem remota”, desenvolvido por Chimenes (2017), baseou-se na criação de um *drone* e uma GCS para auxiliar o piloto durante o voo, exibindo informações sobre a aeronave. Embora já existam diversas soluções de telemetria e softwares GCSs de código-fonte abertos, o autor preferiu desenvolver suas próprias soluções, pelo fato de poder explorar mais a fundo essas tecnologias, ao contrário de somente utilizá-las e também para obter uma solução mais simples e específica.

O quarto trabalho, “Implementação de um *quadrotor* como plataforma de desenvolvimento para algoritmos de controle”, elaborado por Melo (2010), também busca a utilização do uso de sensores para a tomada de decisões, mas em um contexto diferente. O autor não só montou um *drone*, mas também desenvolveu a placa controladora e o *firmware*, com o intuito de que o mesmo sirva como base para o desenvolvimento de futuros trabalhos, focados em algoritmos de estabilização de voo baseados nos sensores a bordo e autonomia de decisões, um *drone* como plataforma de desenvolvimento.

Alguns dos autores utilizaram placas controladoras existentes no mercado, enquanto outros projetaram a mesma. O mesmo ocorre para o software, que alguns autores

desenvolveram o próprio. A Tabela 5 apresenta um resumo das tecnologias utilizadas pelos autores.

Tabela 5 – Resumo das tecnologias utilizadas nos trabalhos relacionados

Áreas	Almeida (2015)	Barato (2014)	Chimenes (2017)	Melo (2010)
Placa controladora	APM 2.6	Projetada pelo autor	Naza-M Lite	Projetada pelo autor
Software GCS	Mission Planner modificado	Não utilizou	Desenvolvido pelo autor	Não utilizou
<i>Frame</i>	<i>Hexacopter</i>	Não utilizou	<i>Quadcopter</i>	<i>Quadcopter</i>
Transmissão de vídeo	Não	Não	Sim	Não

Fonte: Elaborado pelo autor (2018).

Através dos trabalhos relacionados apresentados é possível perceber que nos últimos anos houve um grande crescimento nas aplicações com uso de VANTS e que já existem muitos estudos sendo realizados nesta área, porém ainda existe bastante a ser explorado e melhorado. Algumas ideias propostas nos trabalhos realizados pelos autores servem como estímulo para que outros autores sejam instigados a dar continuidade ou mesmo usá-las como inspiração para propor novas melhorias. Além disso, os trabalhos apresentados são fontes de referencial teórico para compreender as tecnologias envolvidas.

Neste capítulo foram apresentados os principais conceitos, componentes e softwares necessários para o desenvolvimento do projeto proposto, juntamente com um estudo de trabalhos relacionados. No próximo capítulo será apresentada a metodologia adotada no trabalho, as atividades desenvolvidas e também a forma de avaliação dos resultados.

3 METODOLOGIA

No capítulo anterior foram apresentados os principais conceitos, componentes e softwares necessários para o desenvolvimento do projeto proposto, juntamente com um estudo de trabalhos relacionados. Neste capítulo é apresentada a metodologia adotada no trabalho, as atividades desenvolvidas e a forma de avaliação dos resultados.

Segundo Marconi e Lakatos (2003), a utilização de métodos científicos não é de uso restrito da ciência, porém inexiste ciência sem a aplicação de métodos científicos. Método é um grupo de práticas sistemáticas e racionais que permitem alcançar os objetivos da pesquisa, com maior segurança e economia. É um conjunto de conhecimentos que auxiliam o cientista a traçar o caminho a seguir, a detectar erros no decorrer da pesquisa e a tomar as decisões necessárias.

3.1 Delineamento

No que diz respeito aos objetivos gerais, essa pesquisa pode ser classificada como uma pesquisa exploratória, que segundo Gil (2010), possui o propósito de prover maior familiaridade com o problema, tornando-o mais claro. O planejamento de pesquisas exploratórias é muito flexível, pois vale considerar os mais diversos aspectos pertinentes ao fato ou fenômeno estudado.

Para alcançar os objetivos deste trabalho, foi realizada uma análise das tecnologias disponíveis no mercado para o desenvolvimento do *drone*, através da pesquisa de referencial teórico e estudo de trabalhos relacionados. Por tanto, no que diz respeito aos procedimentos técnicos, esta pesquisa pode ser caracterizada como uma pesquisa bibliográfica, que segundo Gil (2010), é elaborada baseando-se em materiais já publicados. O referencial teórico possui o intuito de prover fundamentação teórica ao trabalho, identificando também o estágio do

conhecimento relacionado ao tema.

Em seguida, é realizado um experimento prático onde é desenvolvido o protótipo de um *drone* e o software para monitoramento e controle do mesmo, demonstrando o funcionamento da proposta. Dessa forma, essa pesquisa também pode ser caracterizada quanto aos procedimentos técnicos, como uma pesquisa do tipo experimental, que segundo Gil (2010), consiste na determinação de um objeto de estudo, manipulando suas variáveis, controlando e observando os efeitos que estas variáveis fazem no objeto. O pesquisador é um agente ativo, ao contrário de um observador passivo.

3.2 Cenário

O cenário do experimento é uma área de terras onde existem sensores, que podem ser de fumaça para detectar incêndios, de nível da água para detectar alagamentos, de calor, de humidade, entre outros. As coordenadas dos sensores são previamente conhecidas e ao serem acionados disparam alertas para um usuário monitor da área, informando que algum incidente aconteceu. O usuário recebe um aviso no qual é solicitada a permissão para o envio do *drone*. Caso a permissão for concedida, o drone sobrevoa o local do sensor de forma autônoma, independente do usuário, transmitindo imagens e permitindo ao monitor comprovar a validação do incidente, podendo tomar as medidas necessárias.

As aproximações do *drone* aos locais dos sensores ocorrem através de missões pré-definidas, geradas de forma manual através de um software planejador de missões, onde são definidos os caminhos que o *drone* percorre e as coordenadas onde o mesmo deve capturar e transmitir imagens.

3.3 Protótipos

No que refere-se a hardware, este trabalho contempla a construção do protótipo do *drone*, montado a partir de componentes adquiridos. De forma geral, o *drone* é montado em um *frame* do tipo *quadcopter*, com uma placa controladora de voo que permite voos autônomos através de dados recebidos dos sensores de giroscópio, barômetro, bússola e GPS, além de outros componentes eletrônicos necessários.

No que diz respeito a software, é desenvolvida uma Estação de Controle Terrestre (GCS), com capacidade de monitorar a posição do *drone*, autorizar a execução de missões e

visualizar as imagens transmitidas pelo mesmo.

3.4 Avaliação da proposta

Para a avaliação da solução proposta, são realizadas simulações virtuais e aplicações reais, nas quais são disparados alertas de sensores pré-definidos, seguidas de permissões para o *drone* sobrevoar o local. Os pontos são previamente cadastrados e conhecidos. Enquanto o *drone* decola e sobrevoa o local, as imagens transmitidas são visualizadas, verificando se o mesmo desloca-se para as coordenadas corretas. Posteriormente, é verificado se o *drone* retorna ao local de lançamento e realiza o pouso.

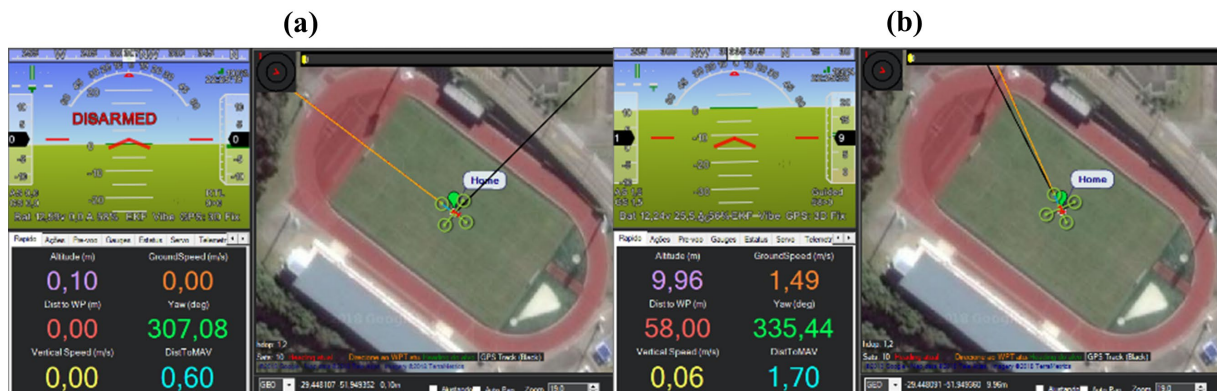
4 DESENVOLVIMENTO

Este capítulo contempla o desenvolvimento dos protótipos, abordando as escolhas dos componentes, a montagem do *drone*, o desenvolvimento do software da Estação de Controle Terrestre (GCS – *Ground Control Station*), os testes com simulador, os testes em uma aplicação prática e a avaliação dos resultados.

4.1 Simulações de missões de voo

No decorrer da primeira parte deste trabalho, viu-se a necessidade de realizar alguns testes com o objetivo de verificar a viabilidade de comunicação de softwares da Estação de Controle Terrestre com o *drone*, podendo validar a possibilidade de realização do experimento prático proposto. Como experimento, foi simulado com o software SITL um *drone* do tipo *quadcopter*, iniciado em coordenadas geográficas de teste. Em seguida, foi estabelecida a comunicação com o mesmo via MAVLink, para o envio de comandos e para o monitoramento de missões com o Mission Planner (MP). A Figura 25 contém duas capturas de tela do MP.

Figura 25 – Decolagem do *drone*



Fonte: Do autor (2018).

Os comandos foram enviados através de *scripts* desenvolvidos com a API DroneKit, sendo o principal teste uma simulação de missão, iniciada com o envio de um comando de acionamento de motores e decolagem para a altura de 10 m. A imagem (a) da Figura 25 demonstra o *drone* aguardando comandos com os motores desarmados e a imagem (b) demonstra o *drone* após a decolagem, a uma altitude aproximada de 10 m. Uma vez que o *drone* simulado alcançou a altitude solicitada, foram enviadas coordenadas geográficas para as quais o mesmo deveria se posicionar. Foram definidas 4 coordenadas, representadas na Tabela 6.

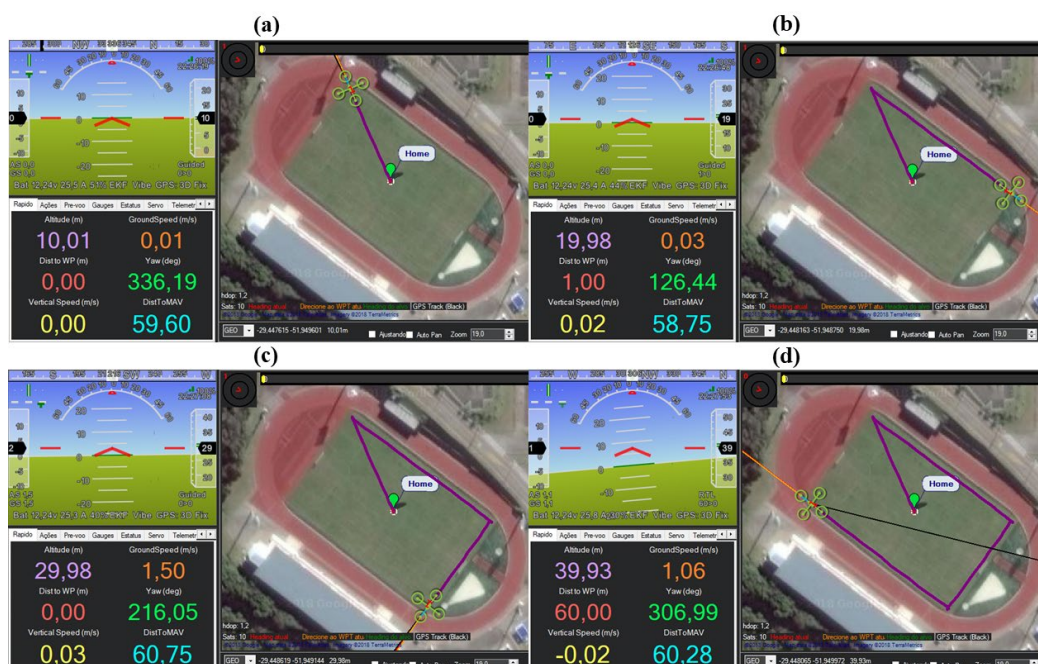
Tabela 6 – Coordenadas geográficas da missão

Coordenada	Latitude	Longitude	Altura
Coordenada inicial	-29.448103	-51.949357	0 m
Coordenada 1	-29.447612	-51.949602	10 m
Coordenada 2	-29.448157	-51.948759	20 m
Coordenada 3	-29.448618	-51.949142	30 m
Coordenada 4	-29.448064	-51.949977	40 m

Fonte: Elaborado pelo autor (2018).

As coordenadas foram enviadas sequencialmente, sempre aguardando o *drone* atingir uma coordenada para enviar a próxima. O voo foi observado através do Mission Planner (MP) e a Figura 26 apresenta quatro capturas de tela, sendo que cada uma ilustra o momento em que o *drone* atingiu uma das coordenadas.

Figura 26 – *Drone* executando uma missão



Fonte: Do autor (2018).

As imagens (a), (b), (c) e (d) da Figura 26 demonstram a chegada do *drone* a cada uma das coordenadas. Depois que o *drone* chegou na última coordenada, foi ativado o modo *return to launch*, no qual o *drone* volta automaticamente para as coordenadas de partida ou casa (sinalizado como “Home” no MP) e realiza o pouso. As coordenadas de partida são definidas automaticamente pelo *drone*, depois que o mesmo é ligado e os satélites GPS forem descobertos, mas no caso deste teste as coordenadas foram definidas na execução do simulador. A Figura 27 é uma captura de tela do MP no momento em que o *drone* aterrissa e encontra-se desarmando os motores, concluindo assim a missão.

Figura 27 – Momento em que o *drone* retorna à coordenada inicial.



Fonte: Do autor (2018).

Além desta missão, foram executados diversos outros testes com o auxílio do simulador, como decolagem seguida de aterrissagem, deslocamento de um ponto a outro, troca de coordenada alvo durante o deslocamento e troca de modos de voo.

Os resultados foram satisfatórios, uma vez que é possível desenvolver um software para monitoramento e controle do *drone* a partir da DroneKit, uma API bem documentada e que já contempla diversas funções prontas para serem utilizadas. Dessa forma pôde ser dada continuidade ao desenvolvimento do trabalho, definindo quais componentes utilizar para a montagem do *drone*.

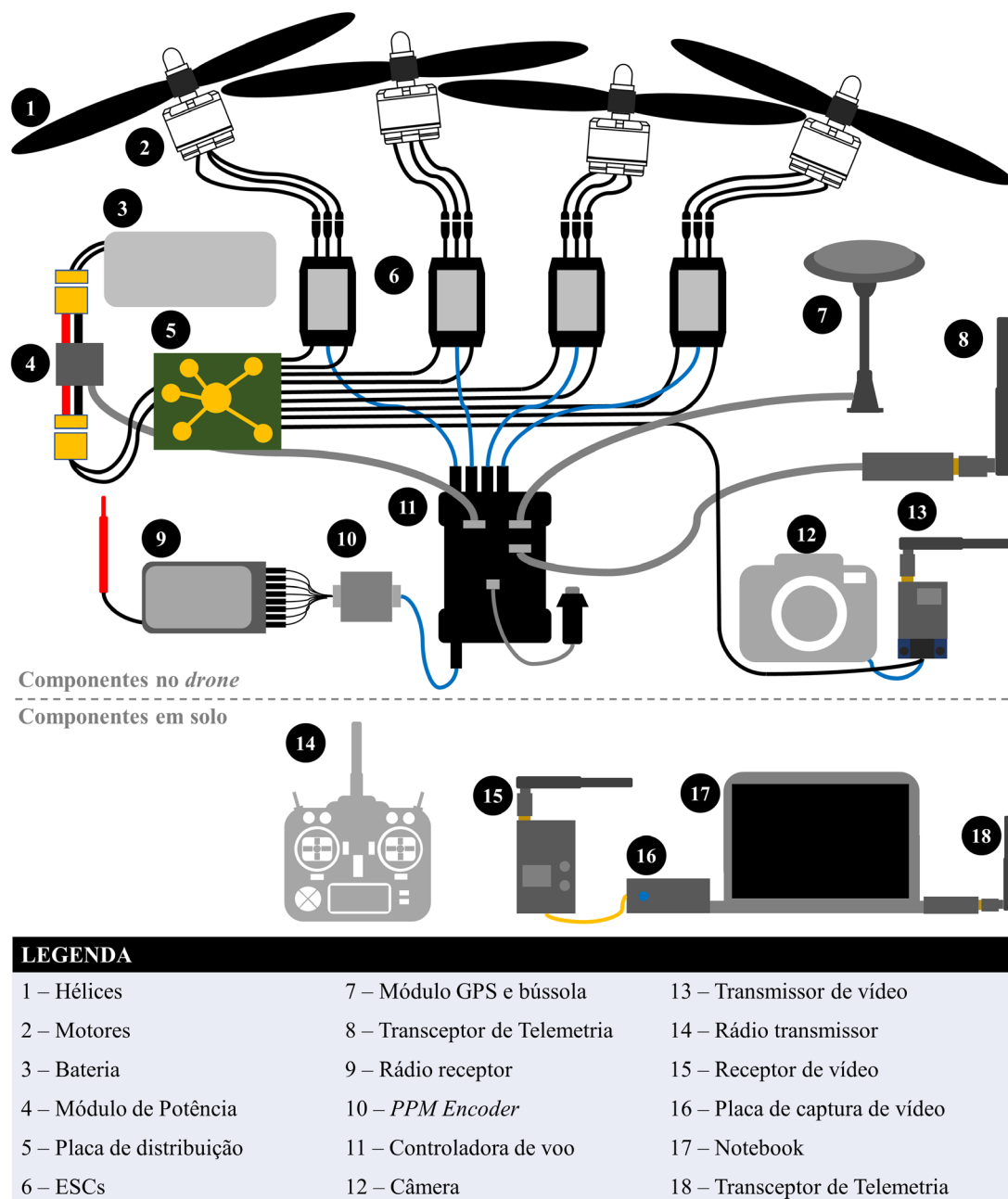
4.2 Desenvolvimento do protótipo do *drone*

Drones do tipo *quadcopter* possuem menos componentes eletrônicos, se comparados a *drones* do tipo *hexacopter* ou *octocopter*, devido a quantidade de conjuntos de hélices e motores. Por estes motivos, e por um menor custo, optou-se por montar o protótipo baseado em um *frame* do tipo *quadcopter*.

A bordo do *drone* há somente o necessário para o mesmo voar executando as missões que lhe são definidas, utilizando os sensores para manter-se estável, determinar seu posicionamento, comunicar-se com a estação terrestre, capturar e transmitir imagens.

A estação terrestre é constituída de um notebook, com os componentes necessários para monitorar e comunicar-se com o *drone*, autorizar a execução de missões e visualizar as imagens transmitidas pelo mesmo. A Figura 28 ilustra todos os componentes necessários para a montagem do protótipo, além de exemplificar a conexão entre os mesmos.

Figura 28 – Componentes do protótipo e esquema de conexões



Fonte: Elaborado pelo autor (2018).

A próxima seção aborda cada um dos componentes representados na Figura 28, detalhando as características e os motivos para a escolha dos mesmos.

4.2.1 Componentes utilizados para a montagem

O *frame* escolhido foi o modelo S500, devido ao seu tamanho de 500 mm de diâmetro, que permite acoplar todos os componentes necessários, com a possibilidade de adição de outros que possam ser necessários no decorrer dos experimentos. Um detalhe importante na escolha do *frame* foi o seu trem de pouso que possui uma altura de 150 mm, deixando toda a eletrônica longe do solo, quando em repouso. Além disso o *frame* é configurado em formato de “x”, que é interessante para o caso de ser necessário apontar uma câmera para a frente. A Figura 29 é uma foto do *frame* adquirido.

Figura 29 – *Frame* S500



Fonte: Do autor (2018).

Segundo o manual que acompanha o *frame*, os motores indicados para o mesmo são motores modelo 2212 de 980 kV, 2216 de 900 kV ou 3108 de 900 kV. As hélices indicadas são 1045, 1047, 1147 ou 1238. As baterias sugeridas são de 3 ou 4 células, variando de 2200 mAh até 5200 mAh. A partir destas indicações, foram escolhidos os motores do modelo 2212 de 980 kV (FIGURA 30), principalmente pelo fato destes serem bastante comuns no mercado. A medida de kV é igual a quantidade de RPM multiplicados por volts (V), isso quer dizer que para cada 1 V aplicado no motor escolhido, o mesmo terá 980 RPM, ou seja, aplicando-se 11.1 V (tensão nominal de uma bateria 3 S) o mesmo terá aproximadamente 10.878 RPM.

Figura 30 – Motores 2212 980 kV



Fonte: Do autor (2018). (a) Motor de sentido anti-horário e (b) motor de sentido horário.

Outra variável importante para a escolha deste motor é que o mesmo possui maior rotação do que um motor de 900 kV, o que garante maior velocidade e agilidade no ar, com um pouco menos de torque. Este fator também influencia na escolha dos tamanhos das hélices, pois para motores de maiores rotações são indicadas hélices menores, uma vez que hélices grandes são indicadas para motores de baixa rotação, com objetivo de obter maior força de empuxo.

Cada motor pesa aproximadamente 50 g e trabalha a uma tensão de 11.1 V, podendo ser alimentado por baterias LiPo de duas a quatro células. O fabricante recomenda a utilização de dois modelos de hélices, 8045 ou 1045, sendo que, com hélices 1045 o mesmo consome uma corrente de 10.6 A, podendo elevar até 710 g. Quando utilizado com hélices 8045, o mesmo consome uma corrente de 8.1 A, podendo elevar 535 g. Pensando em garantir uma maior força de empuxo, levando a bordo todos os componentes necessários e ainda por estar entre as indicações de hélices do *frame*, foram escolhidas as hélices do modelo 1045 que, em conjunto com os motores escolhidos, permitem decolar com até 2840 g, resultado de 710 g multiplicadas por 4 (número de motores). Assim como os motores, são necessários dois pares de hélices para um *drone* do tipo *quadcopter*, um par de sentido anti-horário e outro par de sentido horário. A Figura 31 representa os dois tipos de hélices utilizadas.

Figura 31 – Hélices 1045



Fonte: Do autor (2018). (a) Hélice de sentido anti-horário e (b) hélice de sentido horário.

A nomenclatura do modelo da hélice 1045 representa o tamanho e o passo da hélice, sendo o seu diâmetro de 10 polegadas e seu passo (ver Seção 2.6.7) de 4,5 polegadas, ou seja, a cada volta completa a distância percorrida dentro do giro é de 4,5 polegadas.

Como visto anteriormente, a bateria indicada para o *frame* escolhido deve ser de 2200 mAh até 5200 mAh. Baseado nisso foi escolhida uma bateria LiPo de 4200 mAh de 3 células, por possuir maior capacidade que a de 2200 mAh e por ser mais leve que a de 5200 mAh. Esta bateria tem a capacidade de fornecer 3.7 V nominais por célula, ou seja, um total de 11.1 V nominais. A mesma pode chegar a 4.2 V por células ou um total de 12.6 V quando totalmente carregada. Possui uma taxa de descarga de 30 C, ou seja, sua corrente nominal de 4200 mAh pode chegar a picos de até 126.000 mAh. A Figura 32 é uma foto da bateria adquirida.

Figura 32 – Bateria LiPo 4200 mAh



Fonte: Do autor (2018).

Cada motor, com a hélice escolhida, opera a uma corrente nominal de 10.6 A e em momentos máximos de pico a corrente fica abaixo de 30 A, portanto ESCs que trabalham a essa corrente são suficientes. Sendo assim, foram escolhidos ESCs do modelo OPTO 30 A da marca RCTimer, indicados pelo fabricante para uso em *drones*. Estes ESCs consomem uma corrente de 30 A, podendo chegar a picos de 40 A e permitem tensões de entrada que variam de 6 V à 16.8 V com baterias de 2 a 4 células. A Figura 33 exibe um dos ESCs adquiridos.

Figura 33 – ESC OPTO 30 A

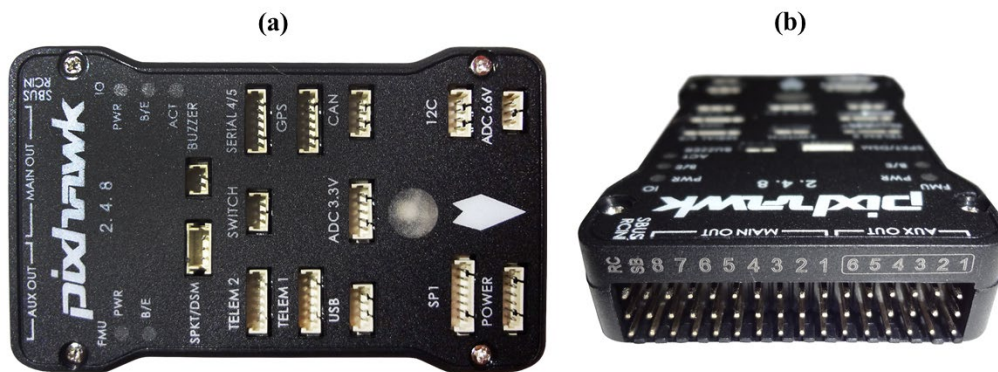


Fonte: Do autor (2018).

Outro motivo que levou a escolha deste modelo de ESC é que o mesmo possui um acoplador óptico, que consiste em um circuito eletrônico que possui um emissor e um receptor de feixes de luz. Dessa forma não existe uma ligação elétrica entre o circuito do ESC e da placa controladora, evitando assim que ruídos gerados pelos motores sejam transmitidos para a mesma.

A controladora de voo escolhida foi a Pixhawk, principalmente por ela permitir voos autônomos, por ser compatível com o *firmware* Copter do Ardupilot e por estar, até a data do presente trabalho, recebendo novas atualizações. A possibilidade de utilização da placa APM foi descartada, uma vez que ela foi descontinuada e não recebe novas atualizações do *firmware*. Foi adquirida uma controladora da versão 2.4.8 do primeiro modelo da Pixhawk, porém já existe no mercado uma segunda versão, com componentes mais modernos e com um valor de custo mais elevado. A Figura 34 apresenta duas fotos da controladora.

Figura 34 – Controladora Pixhawk



Fonte: Do autor (2018). (a) Vista superior da controladora e (b) vista das conexões laterais da controladora.

A Pixhawk vem acompanhada de um interruptor de segurança (FIGURA 35), que serve para ativar ou desativar as saídas para os motores. Dessa forma, só é possível armar o *drone* após o interruptor ser acionado, evitando de armar os motores acidentalmente antes disso.

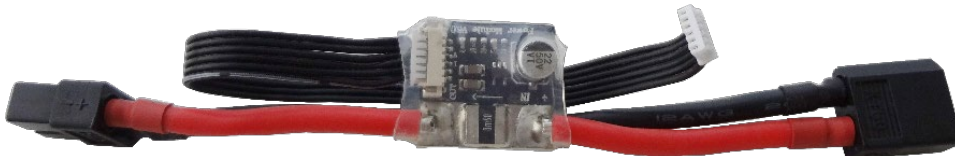
Figura 35 – Interruptor de segurança



Fonte: Do autor (2018).

A Pixhawk precisa ser alimentada por uma fonte de energia estável, para isso foi adquirido um componente chamado Módulo de Potência (FIGURA 36), um componente projetado para a Pixhawk que fornece uma alimentação estável de 5.37 V e 2.25 A. O módulo permite que a controladora monitore a tensão e a corrente da bateria, podendo executar *failsafe* de bateria, que faz com que o *drone* retorne ao local de lançamento quando a tensão ficar muito baixa ou se a energia total consumida durante o voo aproximar-se da capacidade da bateria.

Figura 36 – Módulo de Potência



Fonte: Do autor (2018).

Uma das extremidades do Módulo de Potência possui um conector para a bateria, aceitando uma tensão de entrada de até 18 V (bateria LiPo de até 4 S), sendo então compatível com a bateria adquirida. Na outra extremidade, possui um conector para alimentar a placa de distribuição do *drone* com a mesma tensão de entrada e um conector para alimentar de forma controlada a Pixhawk.

Um módulo de GPS é um componente fundamental para o *drone* executar missões autônomas, pois é necessário, a cada instante ter conhecimento da sua posição geográfica. Foi então adquirido um módulo GPS Ublox Neo-M8N com bússola. Segundo o fabricante, sua precisão fica entre 0.6 m e 0.9 m, que é muito boa se comparado a um módulo GPS Neo-7N, que fica entre 1.4 m e 1.6 m. A Figura 37 é uma imagem do módulo GPS e sua haste de fixação.

Figura 37 – Módulo GPS com bússola

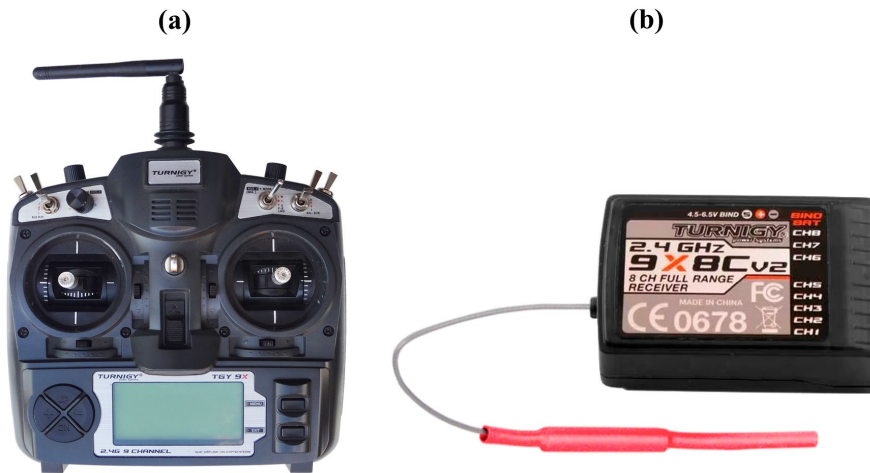


Fonte: Do autor (2018).

Neste trabalho foi utilizado um rádio transmissor somente como fator de segurança, para o caso de ocorrer problemas durante os testes. Para isso também é necessário um rádio receptor

acoplado ao *drone*. O rádio transmissor adquirido foi o Turnigy 9X, que opera a 2.4 Ghz e o rádio receptor foi o Turnigy 9X8C V2, que recebe 8 canais em modo PPM e opera a 2.4 Ghz. O principal motivo para a escolha deste rádio receptor é que ele é muito popular entre a comunidade de aeromodelistas, o que facilita a configuração, além de possuir vários recursos e um custo acessível. A Figura 38 apresenta o rádio transmissor e o rádio receptor utilizados.

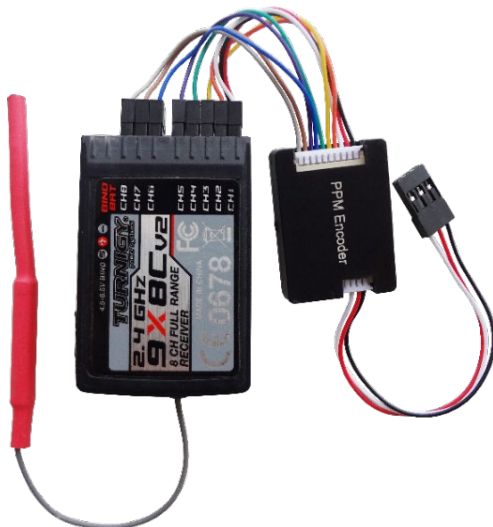
Figura 38 – Rádio transmissor e receptor



Fonte: Do autor (2018). (a) Transmissor e (b) receptor.

O rádio receptor Turnigy 9X8C V2 possui 8 saídas PWM, uma para cada canal. Porém a controladora Pixhawk possui apenas uma entrada do tipo PPM que recebe uma soma de todos os canais de PWM e decodifica internamente. Por este motivo foi necessário adquirir um componente chamado *PPM Encoder*, que permite codificar até 8 sinais PWM em apenas um sinal PPM. A Figura 39 é uma foto do *PPM Encoder* conectado às saídas do rádio receptor.

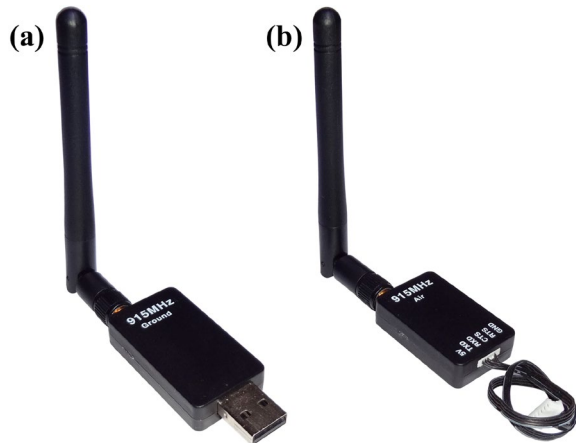
Figura 39 – *PPM Encoder* conectado ao rádio receptor



Fonte: Do autor (2018).

Para estabelecer conexão de telemetria entre o *drone* e a GCS, foram adquiridos transceptores de telemetria 3DR V1 de 915 MHz. Um deles é conectado à placa Pixhawk do *drone* e o outro é conectado a uma porta USB do notebook que atua como GCS. A Figura 40 é uma foto dos transceptores utilizados.

Figura 40 – Transceptores de telemetria



Fonte: Do autor (2018). (a) Transceptor de solo e (b) transceptor de bordo.

Para realizar a captura das imagens foi utilizada uma câmera do tipo *Action Camera* (câmera de ação em português), que foram desenvolvidas para serem utilizadas por esportistas, mas acabaram popularizando-se para os mais diversos públicos. A câmera adquirida foi uma SJ5000X da marca SJCAM, retratada na Figura 41. Ela permite gravar vídeos nas resoluções 1080P a uma taxa de 60 fps (*frames per second*, quadros por segundo), 2 K a uma taxa de 30 fps e 4 K a uma taxa de 24 fps. Além disso, ela possui uma saída de vídeo do tipo AV (Áudio e Vídeo), que foi um dos principais motivos para a escolha da mesma, pois é necessário para a transmissão das imagens para a estação terrestre.

Figura 41 – Câmera SJ5000X

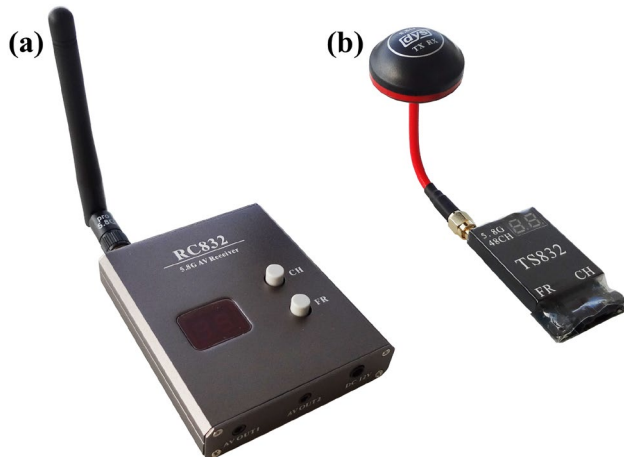


Fonte: Do autor (2018).

As imagens capturadas pela câmera são enviadas através de um transmissor FM de áudio e vídeo, modelo TS832 da marca Eachine e recebidas através de um receptor modelo RC832 também da Eachine. Ambos trabalham na frequência de 5.8 Ghz e a potência do transmissor é

de 600 mW que, segundo o fabricante, permite que as imagens transmitidas chegam a uma distância segura de 5 Km, podendo chegar a 8 Km, dependendo da antena utilizada. Com objetivo de garantir o maior alcance possível, a antena original de 2 dBi¹⁵ do transmissor foi substituída por uma antena de 4 dBi. A Figura 42 é uma imagem do receptor e do transmissor adquiridos.

Figura 42 – Receptor e transmissor de vídeo



Fonte: Do autor (2018). (a) Receptor e (b) transmissor.

Para exibir as imagens recebidas é utilizado um adaptador USB de captura de áudio e vídeo, modelo EasyCap (FIGURA 43), que recebe as imagens analógicas do receptor de vídeo e converte para formato digital, permitindo que as mesmas sejam visualizadas dentro de softwares.

Figura 43 – Adaptador EasyCAP



Fonte: Do fabricante (2018)¹⁶.

¹⁵ É uma unidade de medida utilizada para representar o ganho (potência) de uma antena, com relação a uma antena isotrópica.

¹⁶ Disponível em <https://www.banggood.com/pt/Easycap-USB-2_0-Audio-Video-VHS-VCR-TV-to-DVD-Converter-Capture-Card-Adapter-p-1026608.html>. Acesso em 26 set. 2018.

4.2.2 Custos

Todos os componentes adquiridos, exceto o notebook utilizado como estação terrestre, foram adquiridos através de sites de compras chineses e não foram aplicadas taxas de importação sobre os mesmos. A Tabela 7 apresenta os custos envolvidos com a compra dos componentes utilizados neste trabalho, sendo que estes podem sofrer alterações devido à variação do dólar e a mudança dos preços praticados pelos fornecedores.

Tabela 7 – Custos dos componentes

Componente	Custo
Adaptador EasyCap	R\$ 55,00
Bateria	R\$ 118,31
Câmera	R\$ 406,00
Conectores, cabos e velcro	R\$ 41,92
Controladora, módulo GPS e telemetria	R\$ 358,40
ESCs	R\$ 131,40
<i>Frame</i>	R\$ 96,75
Hélices	R\$ 21,51
Motores	R\$ 89,49
Rádio transmissor e receptor	R\$ 325,00
Recarregador de baterias	R\$ 131,53
Receptor de vídeo com bateria	R\$ 101,39
Transmissor de vídeo	R\$ 45,00
Valor total	R\$ 1921,70

Fonte: Elaborado pelo autor (2018).

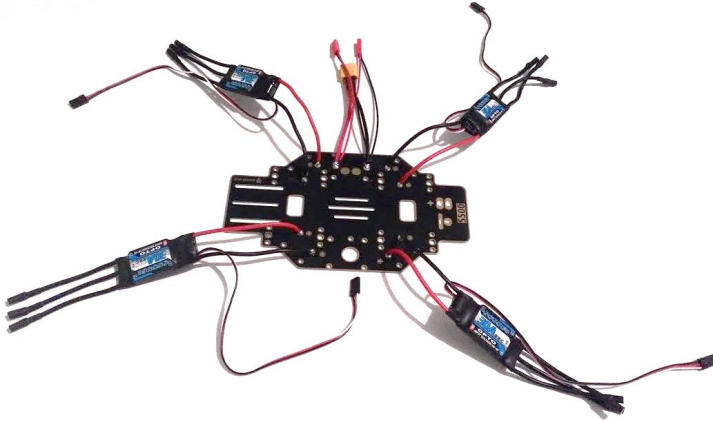
A Tabela 7 não contempla componentes reservas ou de substituição por eventuais acidentes sofridos pelo protótipo. Uma vez que todos os componentes adquiridos estavam disponíveis, foi possível iniciar a montagem do protótipo, assunto da próxima seção.

4.2.3 Montagem

A montagem do *drone* iniciou-se pela soldagem dos quatro ESCs à placa de distribuição de energia do *frame*, garantindo a conexão entre estas peças através de uma liga metálica de estanho. Também foram soldados à placa de distribuição, um cabo com conector XT60 para conexão da bateria LiPo e dois cabos com conectores JST para alimentação do sistema de transmissão de imagens. Nos três terminais de saída de cada ESC foram soldados conectores

de cobre para posterior conexão com os motores, cobertos com isolantes termo retráteis. A Figura 44 demonstra a placa de distribuição já soldada, com os ESCs e com os cabos de entrada e saída de energia.

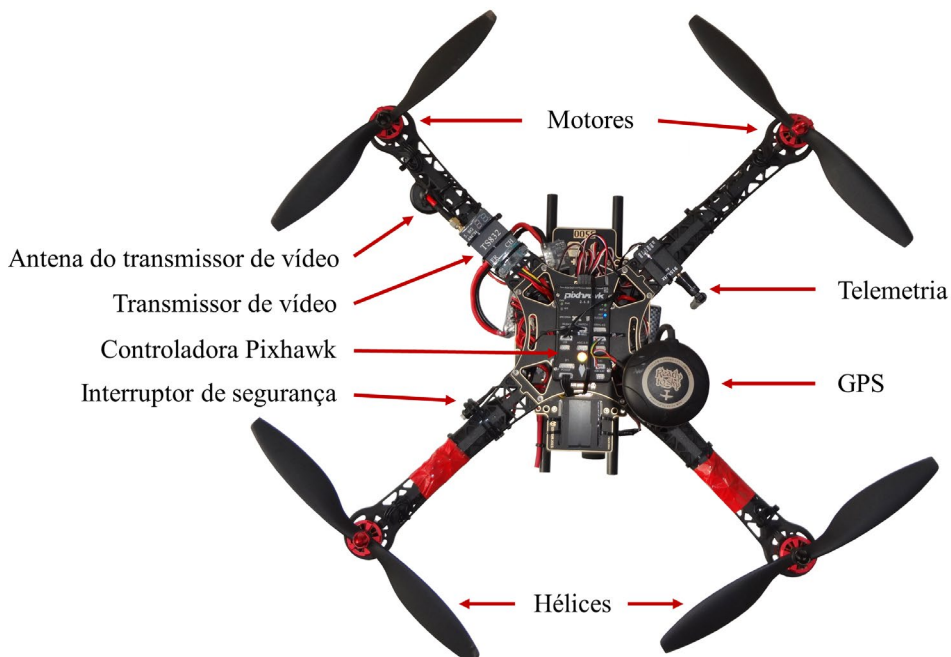
Figura 44 – Componentes soldados à placa de distribuição



Fonte: Do autor (2018).

Em seguida o restante do *frame* foi montado, sendo parafusado na placa de distribuição, que também possui a função de unir todas as partes do *frame*. Na parte superior foi fixada uma outra placa semelhante a placa de distribuição, mas que possui função de servir de suporte ao *frame* e para fixação dos demais componentes. Os ESCs foram fixados nos braços do *frame* através de abraçadeiras plásticas e conectados às saídas dos ESCs. A Figura 45 demonstra os motores fixados nas extremidades dos braços do *frame*.

Figura 45 – Vista superior do drone montado

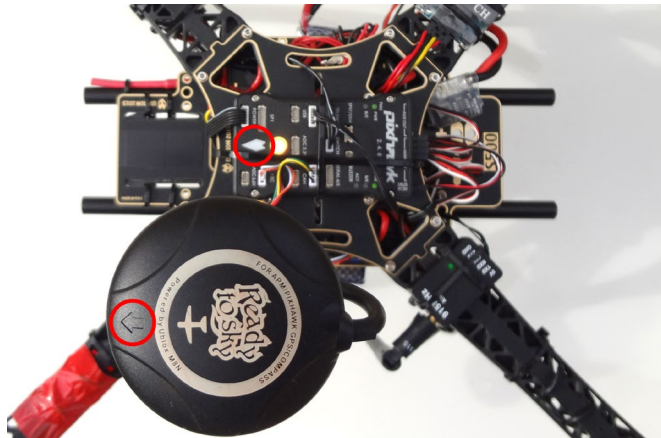


Fonte: Do autor (2018).

Abaixo da placa de distribuição do *frame* foi encaixado o suporte para alojamento da bateria e no braço dianteiro direito foi fixado o interruptor de segurança, ambos com o uso de abraçadeiras plásticas. Sobre a placa de distribuição do *frame*, foi fixado o Módulo de Potência que é conectado à bateria e à placa controladora. Também foi fixado o rádio receptor, conectado ao *PPM Encoder* que por sua vez foi conectado a placa controladora.

A controladora de voo foi fixada ao centro da parte superior do *frame*, recebendo um cabo de alimentação do Módulo de Potência, um cabo do *PPM Encoder*, um cabo do interruptor de segurança, um cabo de cada um dos quatro ESCs, um cabo do módulo de telemetria e um cabo do módulo de GPS, que foi fixado próximo à mesma. Neste ponto é importante ressaltar que o *drone* tem uma frente, para onde a Pixhawk e o módulo de GPS devem apontar, isso serve para a Pixhawk e para a bússola (dentro do sensor de GPS) saberem para qual direção o *drone* está voltado. A Figura 46 demonstra o direcionamento do módulo de GPS e da placa Pixhawk para a frente do *drone*.

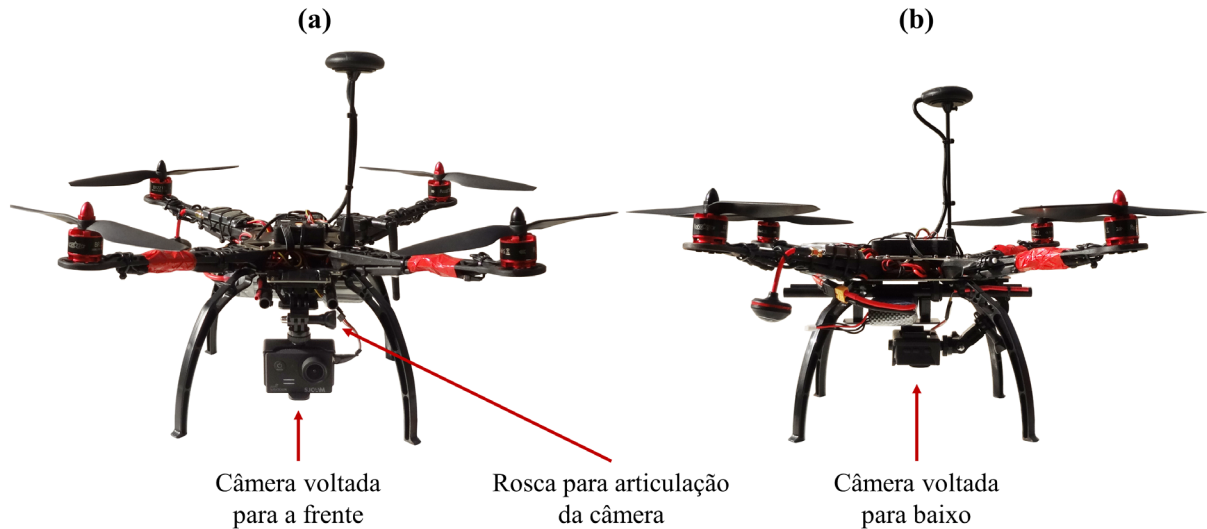
Figura 46 – Direção da bússola e da Pixhawk



Fonte: Do autor (2018). Contornado em vermelho as setas de direcionamento dos componentes.

No braço traseiro esquerdo do *frame* foi fixado o módulo de telemetria e no braço traseiro direito do *frame* foi fixado o módulo transmissor de vídeo, com sua antena virada para baixo, impedindo que a mesma colida com a hélice mais próxima. O transmissor recebe energia através de um dos cabos com conector JST da placa de distribuição e é conectado com a câmera através de um cabo conectado à porta USB da mesma. A câmera foi fixada na parte frontal do *frame*, com a possibilidade de ser articulada para a frente ou para baixo, bastando afrouxar uma rosca. A Figura 47 é uma foto do *drone* montado, onde pode ser percebida a possibilidade de articulação da câmera.

Figura 47 – *Drone* montado



Fonte: Do autor (2018). (a) Vista frontal do *drone* e (b) vista lateral do *drone*.

A massa do *drone* montado ficou em 1510 g. Sua capacidade total é de 2840 g, o que permite, em teoria, decolagens e voos bastante rápidos, e ainda a possibilidade de acoplamento de outros sensores e componentes necessários.

A Estação de Controle Terrestre, utiliza duas portas USB do notebook, em uma é conectado o módulo de telemetria e na outra, é conectado o adaptador de captura de vídeo. Uma bateria 3 S é conectada ao receptor de vídeo que é conectado a placa de captura. A Figura 48 apresenta uma imagem da Estação de Controle Terrestre.

Figura 48 – Estação de Controle Terrestre



Fonte: Do autor (2018).

Antes de tentar qualquer tipo de teste com o *drone*, devem ser realizadas algumas calibrações e configurações, assunto da próxima seção.

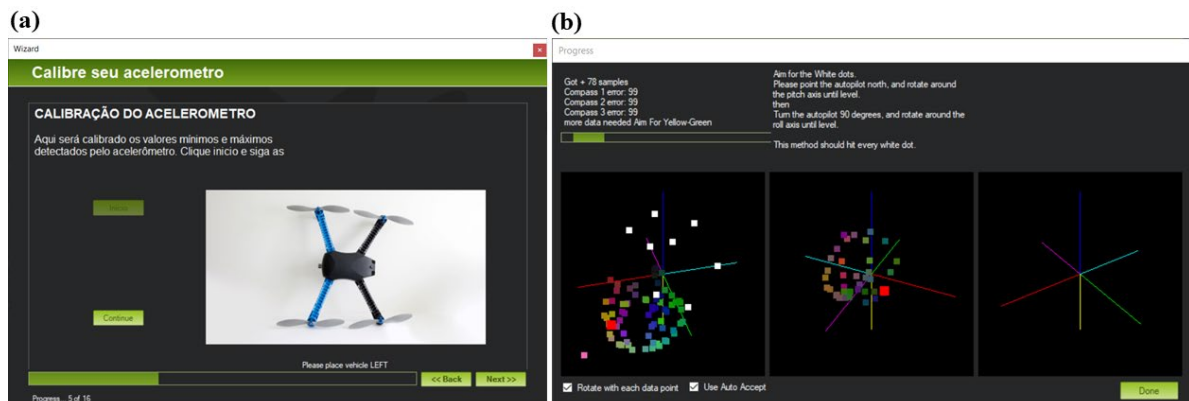
4.2.4 Configurações e calibrações

As calibrações do *drone* atuam sobre a placa controladora de voo e fazem parte da configuração inicial. É necessário configurar alguns componentes do hardware e é possível efetuar a atualização do *firmware* da Pixhawk. Neste trabalho foi utilizado o software Mission Planner (MP) pra fazer todas as configurações e calibrações necessárias no *drone*.

O primeiro passo foi conectar a Pixhawk ao notebook via cabo USB e então, via MP, conectar e executar a atualização para a última versão do *firmware* Copter, versão 3.5.5. O próximo passo foi executar um assistente de configurações do Mission Planner que auxilia na maior parte das configurações e calibrações necessárias, mas que podem ser realizadas separadamente quando for preciso.

O assistente inicia com a escolha do tipo do *frame*, depois passa pela calibração do acelerômetro, onde é necessário posicionar o *drone* de várias formas, apontando para a frente, cima, baixo, direita, esquerda e nivelando o *drone* horizontalmente e de ponta-cabeça. Posteriormente é realizada a calibração da bússola, onde é necessário colocar o *drone* nas mesmas posições que as da calibração do acelerômetro e ainda girá-lo em relação aos eixos x, y e z. Neste passo é necessário estar com *drone* em local externo e aberto, sendo mais cômodo uma conexão com o MP via telemetria, pois via cabo fica mais difícil de girar o *drone*. A Figura 49 demonstra as telas do MP durante a calibração dos acelerômetros e da bússola.

Figura 49 – Telas de calibração do *drone*



Fonte: Do autor (2018). (a) Tela de calibração dos acelerômetros e (b) tela de calibração da bússola.

Em seguida o assistente passa pela configuração dos modos de voo, onde são configurados seis deles, que podem ser selecionados por duas chaves do rádio transmissor. Por último é configurado o *failsafe*, um mecanismo de segurança para a recuperação do *drone*, para casos em que algo der errado durante o voo. Foi configurado o *failsafe* de bateria para 10.5 V

ativando o modo *land*, que faz com que o *drone* pouse quando a bateria baixar desta tensão, visto que possivelmente ocorreu algo de errado e não há bateria suficiente para voar até o local de lançamento e pousar. Também foi configurado o *failsafe* de rádio transmissor, fazendo com que o *drone* retorne para o local de lançamento, caso durante os testes o *drone* perca a conexão com o rádio. Também é possível configurar o *failsafe* para a perda de conexão com a GCS, neste caso foi deixado desabilitado, para que o *drone* continue a missão autonomamente mesmo quando não conseguir mais contato com a GCS, pois a transmissão de vídeo possui um alcance maior, então a GCS continuará a receber as imagens mesmo sem conexão de dados com o *drone*.

4.2.5 Voos de teste

Com o *drone* configurado, foram realizados os testes de voo, onde os primeiros foram realizados controlando o *drone* de forma manual, a partir do rádio transmissor, sendo testados os modos *stabilize*, *altitude hold*, *loiter* e *land*, a fim de verificar a funcionalidade e a capacidade do *drone* manter a estabilidade, altitude e pousar autonomamente.

Figura 50 – Voo manual de teste



Fonte: Do autor (2018). Piloto controlando o *drone* através do rádio transmissor.

Em seguida foram realizados testes com missões autônomas, geradas a partir do Mission Planner e enviadas para o *drone* via telemetria, onde era ativado o modo *auto* e observado o *drone* executando as missões.

No modo *stabilize*, o *drone* manteve-se estável e nivelado, balançando apenas com forças aplicadas pelo vento. O *drone* conseguiu manter a altura no modo *altitude hold*, porém ocorreram algumas variações quando o voo ocorreu em baixas altitudes, pois o sensor

barométrico utilizado apresenta imprecisões em baixas alturas em relação ao solo. Devido essas variações, em uma missão de voo executada a altitude de 2 m, houve uma colisão do *drone* com o solo, tendo como consequência a quebra de duas hélices. No modo *loiter*, que exige a utilização do GPS para manter a posição, foi percebido que o *drone* consegue manter as coordenadas com poucas variações, e no modo *land* o *drone* consegue pousar autonomamente, as vezes tocando bruscamente o solo, mas nada que possa danificar o protótipo.

Durante os testes foi possível perceber algumas vibrações no *drone*, que ficaram mais evidentes quando foram observadas as imagens capturadas pela câmera. Esse problema foi resolvido através do balanceamento de hélices, um procedimento onde ocorre o desgaste no lado mais pesado da hélice, de forma a manter os dois lados com a mesma massa. Para identificar se as hélices estavam desbalanceadas foi utilizado um aparelho chamado balanceador magnético de hélice. A Figura 51 demonstra o balanceamento de uma hélice sendo verificado.

Figura 51 – Balanceamento de hélice sendo verificado



Fonte: Do autor (2018).

O funcionamento do aparelho pode ser comparado com o de uma gangorra, pois o lado da hélice que é mais pesado desce. Estes lados mais pesados foram desgastados com uma lixa de unha fina, no lado que fica para baixo quando instalados no *drone*, de forma uniforme, de modo a não prejudicar o funcionamento e aerodinâmica das mesmas. Foram percebidas melhorias após estes procedimentos.

Com o *drone* montado e operando da forma esperada, foi possível dar início ao desenvolvimento do software de GCS, visto em detalhes na Seção 4.3.

4.3 Desenvolvimento do software

Como apresentado na Seção 4.2.3, o *firmware* que está sendo utilizado na Pixhawk para este trabalho é o Copter, que faz parte da plataforma ArduPilot, escrita praticamente toda na linguagem de programação C++. A fim de facilitar o desenvolvimento dos softwares, optou-

se pela utilização da API DroneKit, que utiliza a linguagem de programação Python, a qual o autor deste trabalho encontra-se mais familiarizado.

4.3.1 Envio de missões para o *drone*

Como pôde ser visto na Seção 4.1, no início da parte prática deste trabalho foram realizados alguns testes iniciais, principalmente como forma de validação da proposta. Posteriormente foram testados vários outros *scripts* Python, com base na documentação do DroneKit¹⁷, que comandavam um *drone* simulado através do software SITL. Todos estes testes resultaram em um *script* Python, com funções de conexão, leitura de arquivo de missões, envio de missões, decolagem e alteração do modo de voo. Antes de enviar qualquer comando para o *drone* é necessário iniciar uma conexão com o mesmo, o código responsável por esta parte é demonstrado na Figura 52.

Figura 52 – *Script* de conexão com o *drone*

```

1 from dronekit import connect, Command, VehicleMode
2 import time
3 import sys
4 import argparse
5
6 vehicle = None
7
8 def connect_to_drone(connection_string):
9     global vehicle
10
11     print 'Conectando-se ao drone...'
12     while not vehicle:
13         try:
14             vehicle = connect(connection_string, wait_ready=True, rate=10)
15         except Exception as e:
16             print 'Aguardando conexao... (%s)' % str(e)
17             time.sleep(2)
18
19     print 'Conectado!'

```

Fonte: Do autor (2018).

É possível perceber através da Figura 52 que na primeira linha do código foram importados alguns módulos da DroneKit e que nas linhas 2, 3 e 4 foram importados alguns módulos do Python. Posteriormente foi declarada a função de conexão com o *drone* que, após ser estabelecida, se mantém acessível através da variável global “vehicle”.

Para auxiliar no envio de uma missão para o *drone*, foi criada uma função chamada “read_mission_file”, que recebe o nome do arquivo de uma missão (gerada através do MP) e retorna uma lista com cada comando da missão, conforme demonstrado na Figura 53.

¹⁷ Disponível em: <<http://python.dronekit.io/develop/index.html>>. Acesso em 12 set. 2018.

Figura 53 – Função responsável pela leitura de um arquivo de missão

```

21 def read_mission_file(file_name):
22     vehicle_commands = vehicle.commands
23     waypoints=[]
24     with open(file_name) as f:
25         for i, row in enumerate(f):
26             if i==0:
27                 if not row.startswith('QGC WPL 110'):
28                     raise Exception('O arquivo nao e compativel com essa versao de WayPoints')
29             else:
30                 row_array=row.split('\t')
31                 ln_index=int(row_array[0])
32                 ln_currentwp=int(row_array[1])
33                 ln_frame=int(row_array[2])
34                 ln_command=int(row_array[3])
35                 ln_param1=float(row_array[4])
36                 ln_param2=float(row_array[5])
37                 ln_param3=float(row_array[6])
38                 ln_param4=float(row_array[7])
39                 ln_param5=float(row_array[8])
40                 ln_param6=float(row_array[9])
41                 ln_param7=float(row_array[10])
42                 ln_autocontinue=int(row_array[11].strip())
43                 command = Command( 0, 0, 0, ln_frame, ln_command, ln_currentwp, ln_autocontinue,
44                                     ln_param1, ln_param2, ln_param3, ln_param4, ln_param5, ln_param6, ln_param7)
45                 waypoints.append(command)
46     return waypoints

```

Fonte: Do autor (2018).

Esta função é utilizada pela função “upload_mission_file”, que é responsável pelo envio da missão para o *drone* e está representada na Figura 54. A função lê a missão que está na memória da controladora do *drone*, apaga todos os comandos dela e em seguida adiciona cada comando da nova missão lida do arquivo. Então é realizado o envio para o *drone*.

Figura 54 – Função que envia as missões para o *drone*

```

47 def upload_mission_file(file_name):
48     waypoints = read_mission_file(file_name)
49
50     vehicle_commands = vehicle.commands
51     vehicle_commands.clear()
52
53     for command in waypoints:
54         vehicle_commands.add(command)
55
56     print('Enviando missao')
57     vehicle_commands.upload()

```

Fonte: Do autor (2018).

Nas versões 3.2.1 ou anteriores do *firmware* Copter não é possível armar o *drone* no modo *auto*, ou seja, iniciar uma missão autônoma com o *drone* em solo. Trata-se de uma questão de segurança que evita que a aeronave seja armada acidentalmente, mas que foi aprimorada nas versões posteriores, a partir do envio de comandos especiais para início de missão em solo. A fim de manter a compatibilidade com a maioria das versões do Copter, foi criada uma função para decolagem em modo *guided*¹⁸, depois disso é possível trocar para o modo *auto*. A Figura 55 apresenta a função de decolagem, chamada de “take_off”.

¹⁸ Modo de voo guiado, no qual o *drone* permanece sendo controlado pela estação de controle terrestre.

Figura 55 – Função de decolagem

```

59 def take_off(altitude):
60     while not vehicle.is_armable:
61         print("Aguardando a inicialização do drone...")
62         time.sleep(1)
63
64     vehicle.mode = VehicleMode("GUIDED")
65     vehicle.armed = True
66
67     while not vehicle.mode.name=="GUIDED" and not vehicle.armed:
68         print "Preparando-se para decolar..."
69         time.sleep(1)
70
71     print("Decolando!")
72     vehicle.simple_takeoff(altitude)
73
74     while True:
75         print("Altura: ", vehicle.location.global_relative_frame.alt)
76
77         if vehicle.location.global_relative_frame.alt >= altitude * 0.95:
78             print("Decolagem completa!")
79             break
80         time.sleep(1)

```

Fonte: Do autor (2018).

Na linha 64 da Figura 55, percebe-se que o modo de voo foi alterado para *guided*, antes do comando de armar da linha 65. Posteriormente, no código há um laço de repetição que impede que o *script* continue até que o *drone* esteja armado e com o modo *guided* ativado. Na linha 72 é executada a função de decolagem da DroneKit para a altura recebida por parâmetro. Depois disso, há um outro laço de repetição que bloqueia o avanço do *script* até que o *drone* atinja a altura desejada.

Para definir quais funções e em qual ordem serão executadas, foram definidos argumentos que devem ser passados durante a inicialização do *script*. Esse trecho do código pode ser visualizado na Figura 56.

Figura 56 – Trecho de inicialização do código

```

82 parser = argparse.ArgumentParser(description='Envio de missoes para o drone')
83 parser.add_argument('--connect', help="Endereco do veiculo.")
84 parser.add_argument('--file', help="Arquivo da missao.")
85 parser.add_argument('--execute', help="Executar a missao.")
86 parser.add_argument('--mode', help="Altera o modo de voo para o modo especificado.")
87 args = parser.parse_args()
88
89 if args.connect:
90     connect_to_drone(args.connect)
91
92 if args.file:
93     upload_mission_file(args.file)
94
95 if args.execute=='1':
96     take_off(10)
97     vehicle.mode = VehicleMode('AUTO')
98
99 if args.mode:
100     vehicle.mode = VehicleMode(args.mode)
101

```

Fonte: Do autor (2018).

Através da Figura 56, percebe-se que foi criado um argumento chamado “connect” que recebe o endereço de rede do *drone*, o argumento “file” que recebe o arquivo da missão, o argumento “execute” que indica se a missão deve ser executada e o argumento “mode” que recebe o modo de voo para qual o *drone* deve ser alterado. É possível passar todos os argumentos ou apenas um por vez.

Na linha 92 do código é verificado se foi passado um arquivo por parâmetro, caso sim será feito o envio da missão do arquivo para o *drone*, na linha 93. Na linha 95 do código é verificado se foi passado o número 1 no argumento “execute”, caso sim, será executada a decolagem para 10 metros (linha 96) e depois é alterado o modo de voo para *auto* (linha 97), então o *drone* vai executar a missão. A Figura 57 é um exemplo de execução do *script* onde é passado o caminho do arquivo de missões e é solicitada a execução.

Figura 57 – Exemplo de execução do *script*

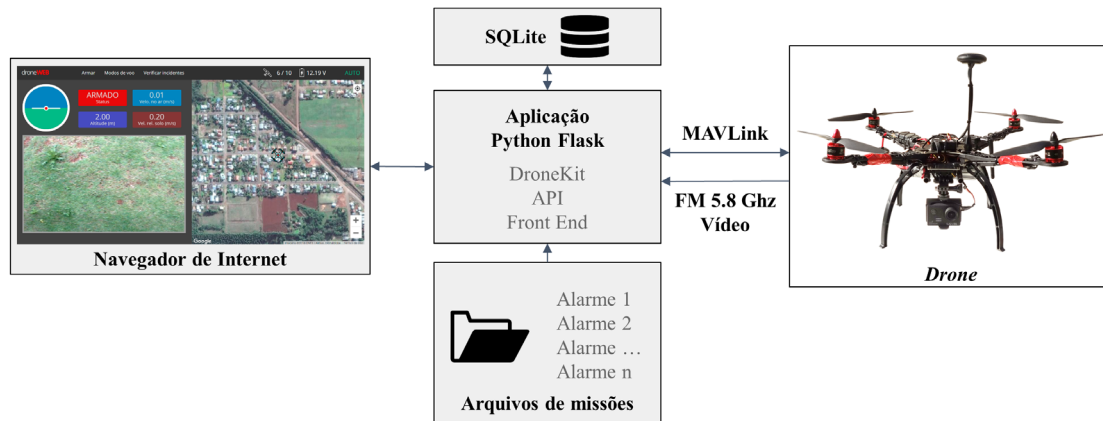
```
diziano@DESKTOP-IUNE243:~/projetos/dronekit$ python send_route_file2.py
--connect=tcp:127.0.0.1:5762 --file=mission_file1.waypoints --execute=1
```

Fonte: Do autor (2018).

4.3.2 Desenvolvimento do software de GCS

O software da Estação de Controle Terrestre foi desenvolvido com a linguagem de programação Python, onde foi adaptado e expandido o *script* DroneKit já desenvolvido, apresentado na Seção 4.3.1 e o *framework* Flask para o desenvolvimento no formato web. Os dados dos alarmes, como coordenadas geográficas, local do arquivo de missão e estado do disparo são persistidos em banco de dados SQLite. Os arquivos das missões de verificação dos alarmes são armazenados em disco. A Figura 58 representa a arquitetura da GCS desenvolvida.

Figura 58 – Arquitetura da GCS



Fonte: Do autor (2018).

O navegador de Internet requisita a aplicação a um servidor web Flask. A camada de visualização e interação da aplicação, também chamada de *front-end*, foi desenvolvida com a linguagem de marcação HTML, a linguagem de folha de estilo CSS e a linguagem de *script* JavaScript com a biblioteca jQuery. A aplicação rodando no navegador realiza requisições à API Flask que comunica com o *drone* através da biblioteca desenvolvida com DroneKit.

As imagens capturadas pelo *drone* são exibidas no navegador através de uma API do JavaScript chamada MediaDevices, que proporciona acesso aos dispositivos de entrada de mídia conectados ao computador, neste caso à placa EasyCap. Os mapas são fornecidos pela plataforma Google Maps, através da API JavaScript, que também fornece a navegação, o controle de zoom e o gerenciamento dos marcadores (locais dos alarmes e do *drone*).

Através do software GCS desenvolvido é possível visualizar a posição no mapa, a inclinação, o status, a altitude, a velocidade e a direção para onde o *drone* aponta. É possível visualizar as imagens capturadas pela câmera, a quantidade de satélites GPS encontrados, a carga da bateria e o modo de voo ativo. Além disso é possível armar os motores, alterar o modo de voo e autorizar o envio do *drone* para a verificação dos alarmes. A Figura 59 é uma captura de tela da aplicação desenvolvida.

Figura 59 – Software GCS desenvolvido



Fonte: Do autor (2018).

Para uma melhor visualização do mapa e dos pontos de verificação, é possível alternar a exibição das imagens, clicando em um ícone de fechar ao passar o mouse por cima do vídeo ou clicando no ícone da câmera quando o vídeo está desativado. A Figura 60 demonstra o vídeo

desativado e alguns pontos de verificação de exemplo.

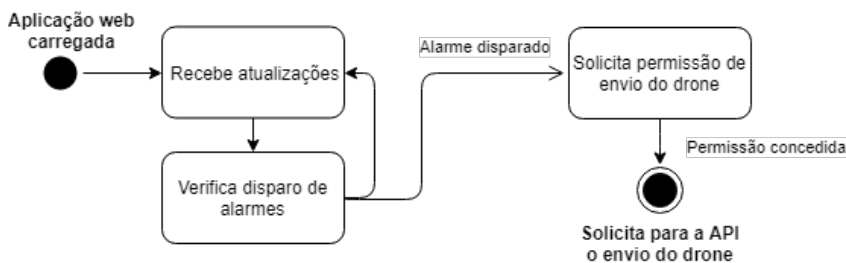
Figura 60 – Visualização dos pontos de verificação



Fonte: Do autor (2018).

O servidor Flask roda uma *thread* que a cada 250 ms envia um evento de atualização com dados do *drone* e dos alarmes. A aplicação rodando no navegador aguarda por estes eventos e a cada recebimento atualiza as informações na tela. Se algum dos alarmes que não estava disparado trocar de status para disparado é soado um alerta sonoro e é exibida uma notificação, solicitando a permissão para o envio do *drone*. Ao ser concedida a permissão, a aplicação envia para o servidor via API uma ordem de verificação do alarme. A Figura 61 é uma representação do envio do *drone* para a verificação de alarme.

Figura 61 – Representação de envio do *drone*



Fonte: Do autor (2018).

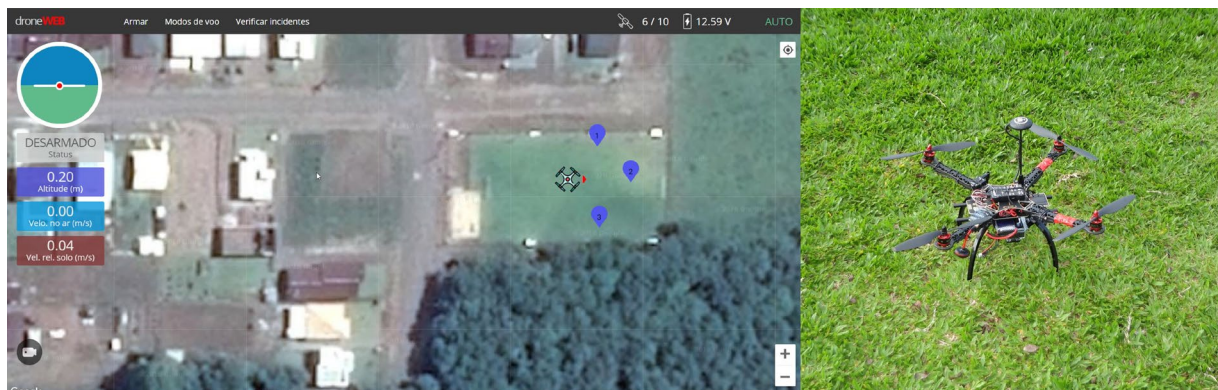
Foi desenvolvida uma tela para a simulação do disparo de alarmes, onde é possível definir alarmes como disparados ou não disparados. Ao salvar os dados são submetidos para a API Flask, que trata de gravar o status do alarme selecionado no banco de dados. Esse recurso facilita os testes, podendo definir rapidamente um alarme como disparado, uma vez que não é necessário possuir alarmes reais.

4.4 Avaliação da proposta

Afim de avaliar a solução proposta, foram realizados testes através de voos reais com o *drone*, porém com alarmes fictícios. O local dos testes foi um campo de futebol, onde três alarmes eram disparados e após a autorização de envio do *drone* era verificado o processo de voo. As missões de voo para cada um dos alarmes foram previamente geradas no software Mission Planner. Esta seção relata uma das verificações de incidentes testadas.

Após a chegada no local dos testes, foi realizada a calibragem da bússola, a fim de garantir uma orientação mais adequada do *drone*. Uma vez que o *drone* obteve a posição GPS, foi apresentado o ícone do *drone* na posição correta, na aplicação GCS desenvolvida. A Figura 62 demonstra o *drone* em solo, aguardando a ocorrência de algum disparo para a verificação.

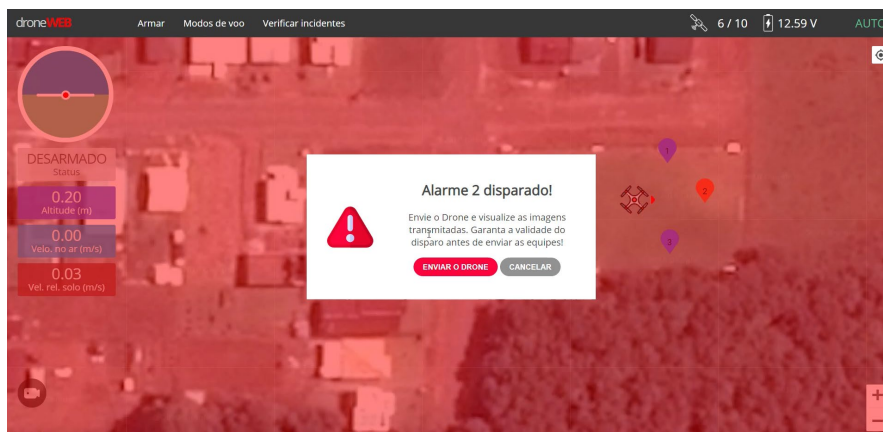
Figura 62 – Drone em solo



Fonte: Do autor (2018).

Simulou-se o disparo de um alarme e conforme pode ser visualizado na Figura 63, a aplicação exibiu uma notificação, solicitando a permissão para envio do *drone*. Também foi soado um alarme sonoro.

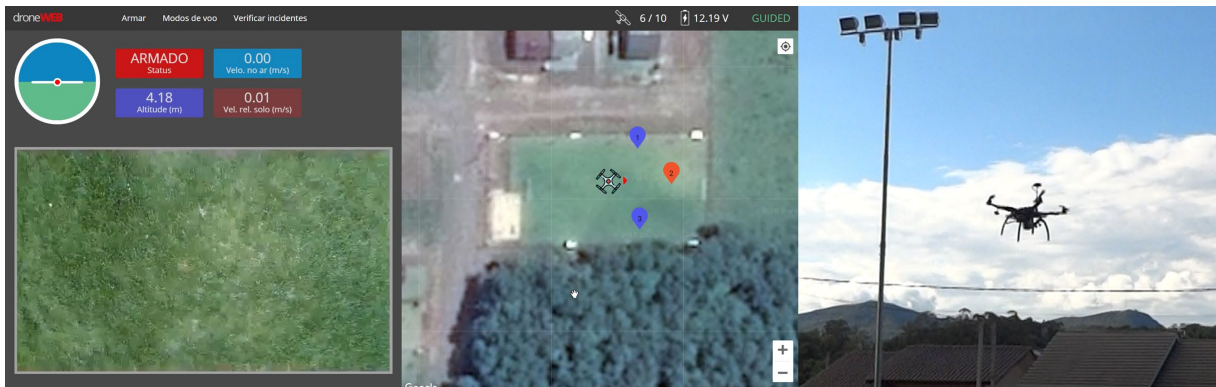
Figura 63 – Notificação de alarme disparado



Fonte: Do autor (2018).

A missão configurada para este alarme era decolar a uma altura de 5 m, voar até o local do disparo, descer a 3 m de altura e permanecer por 5 s. Depois voltar à altura de 5 m e deslocar-se nesta altura até um determinado local (base terrestre, onde o *drone* deve permanecer no aguardo de disparos), onde foi especificado para pousar. Depois de conceder a permissão para a verificação do disparo, os motores do *drone* foram acionados e o mesmo começou a decolar. A Figura 64 demonstra a decolagem do *drone*.

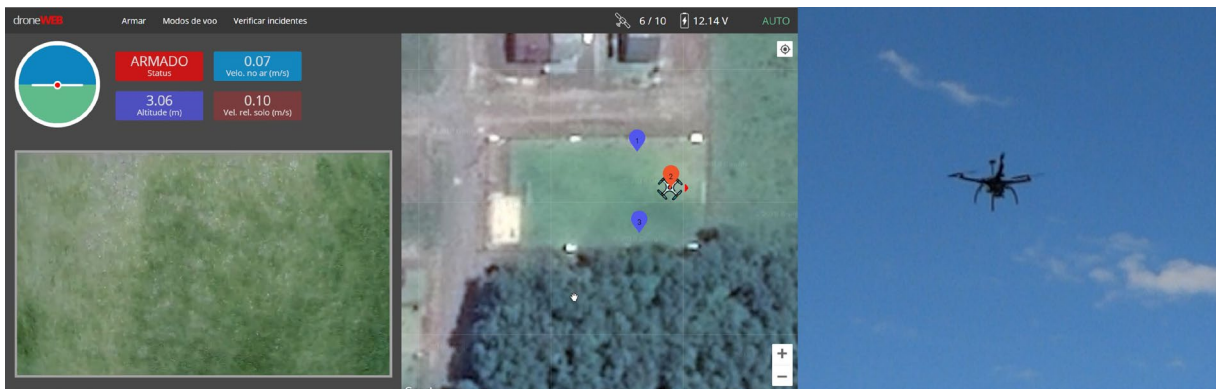
Figura 64 – *Drone* decolando



Fonte: Do autor (2018).

Logo após a decolagem já foi possível perceber o movimento do *drone* em direção ao local do disparo do alarme. Ao chegar no local do disparo do alarme o *drone* desceu aproximadamente 2 m, conforme havia sido especificado na missão, sendo uma forma de aproximar-se do local do incidente, para uma melhor visualização do ocorrido. A Figura 65 demonstra o *drone* sobrevoando o local do disparo, onde ele permaneceu sobrevoando pelo tempo configurado na missão.

Figura 65 – *Drone* sobrevoando o local do alarme



Fonte: Do autor (2018).

Posteriormente o *drone* deslocou-se até o local configurado para o pouso, onde ativou o modo de voo *land*, que o fez aterrissar com sucesso. Dessa forma, a verificação deste incidente

é dada como concluída.

4.5 Resultados

A partir da avaliação da proposta, detalhada na Seção 4.4, é possível concluir que o protótipo do *drone* cumpriu com sucesso as tarefas as quais esperava-se, realizando voos autônomos até as coordenadas dos alarmes estabelecidos, transmitindo imagens do local do incidente, retornando e pousando no local ao qual foi configurado.

O *drone* demonstrou ser ágil na decolagem, no deslocamento e no pouso, além de manter-se estável, quanto a sua altura e posição geográfica. São características que, em modo de voo manual, só seriam atingidas por pilotos muito experientes e dependendo das condições, como local do alarme, alcance da visão do piloto e alcance do controle remoto, poderiam até mesmo ser impraticáveis.

O software GCS também cumpriu o seu papel, uma vez que monitorou o status dos alarmes no banco de dados e notificou o disparo, solicitando o envio do *drone*. O soar do alarme sonoro, junto com a notificação na tela, auxilia na chamada de atenção do usuário que pode estar distraído no momento da ocorrência de algum incidente. A comunicação com o *drone* se mostrou bastante rápida, não apresentando lentidões que prejudicassem a operação, tanto no envio das missões quanto no monitoramento durante o voo.

O trabalho de pesquisa e o desenvolvimento dos protótipos apresentados neste trabalho servem de referencial teórico e prático para futuros trabalhos, uma vez que não será preciso realizar uma nova pesquisa, a fim de verificar os componentes necessários, o funcionamento e o desenvolvimento do equipamento. Este trabalho pode servir de ponto de partida para trabalhos futuros.

4.6 Dificuldades

A maior dificuldade enfrentada no desenvolvimento deste trabalho foi na parte prática, na execução dos testes com o protótipo do *drone*, pois precisavam ser realizados em ambientes externos e espaçosos, sendo também impossibilitados em dias chuvosos. Essa dificuldade foi amenizada com a utilização do software SITL, que é capaz de simular um *drone* com as mesmas características do protótipo desenvolvido e foi utilizado durante todo o desenvolvimento do trabalho. Através do SITL é possível configurar vários parâmetros, como velocidade máxima

do *drone*, tipos de sensores utilizados e até adicionar componentes virtuais, mas para o propósito deste trabalho não foram necessárias configurações adicionais, foram utilizados os parâmetros padrões do simulador para *drones* do tipo *quadcopter*.

Os primeiros testes realizados com o protótipo (Seção 4.2.5) exigiram o comando manual do *drone*, a fim de verificar o seu correto funcionamento. Essa tarefa foi um desafio, pois o autor do trabalho não possuía experiências prévias com pilotagem de *drones*. Esse desafio foi superado com algumas horas de treinamento do autor, através do simulador de voo rádio controlado Phoenix RC 5, onde foi simulado um *drone* que foi comandado através do rádio controle (o mesmo adquirido para o protótipo) conectado ao notebook.

Outra dificuldade encontrada no desenvolvimento do trabalho foi a quebra de peças do *drone* durante os testes reais. Foram quebradas 4 hélices e uma das pernas do trem de pouso. Haviam sido adquiridas hélices reservas e o trem de pouso precisou ser colado e adaptando, sendo utilizado até a chegada de um novo que foi adquirido, evitando impactos no andamento do trabalho.

5 CONCLUSÃO

O presente trabalho teve por finalidade o desenvolvimento de um *drone* com capacidade de voos autônomos para auxílio na verificação de incidentes, com capacidade de auxiliar o trabalho de equipes de verificação. Foi realizada a análise das tecnologias disponíveis no mercado, para *drones* autônomos, através de uma pesquisa de referencial teórico e trabalhos relacionados. Baseado nestes estudos, foi desenvolvido o protótipo de *drone* e o software para monitoramento e controle do mesmo.

O *drone* desenvolvido possui uma boa flexibilidade para modificações, uma vez que o tamanho do *frame* e a relação entre a força proporcionada pelo conjunto, aliado ao peso final do protótipo, possibilitam a adição de mais componentes e sensores, podendo expandir as perspectivas de uso do equipamento. O software desenvolvido também permite melhorias, uma vez que foram utilizadas APIs de código-fonte aberto, desenvolvidas por terceiros e que possuem muitos recursos já desenvolvidos, necessitando de alterações somente em casos muito específicos.

No estudo, ficou evidenciado que é possível utilizar um *drone* para auxílio na validação de disparo de alarmes, mas existem limitações cujas soluções podem ser exploradas em trabalhos futuros. As principais limitações estão relacionadas com o deslocamento de grandes distâncias em relação a estação terrestre, como o alcance do sinal dos módulos de telemetria, o alcance do sinal do sistema de transmissão de vídeo e a carga limitada da bateria. Sugere-se para trabalhos futuros o desenvolvimento de *drones* com formas alternativas de energia como, por exemplo, energia solar. Poderia ser projetado também, um sistema de recarregamento automático da bateria, quando o *drone* encontra-se em solo.

Na Seção 4.2.5 deste trabalho, constatou-se que podem ocorrer imprecisões na altura

medida pelo sensor barométrico em baixas altitudes, o que para o propósito deste trabalho não é um impeditivo do correto funcionamento, uma vez que o *drone* não precisa efetuar grandes aproximações. Apesar disso, sugere-se para trabalhos futuros a adição de outro sensor mais preciso para essas situações, onde o *drone* precisa aproximar-se e manter-se voando próximo ao solo. Existem no mercado sensores do tipo sonar e *rangefinder* (telêmetro em português), os quais são muito precisos para medir curtas distâncias. Sugere-se também a adição de sensores deste tipo ao redor do *drone*, permitindo-o evitar colisões com obstáculos.

Durante os testes realizados, a câmera ficou voltada para baixo, possibilitando a captura de imagens abaixo do *drone*. Acredita-se que as áreas poderiam ser melhor visualizadas se a câmera pudesse girar, fornecendo uma visão perspectiva dos locais dos incidentes. Essa funcionalidade pode ser explorada em trabalhos futuros, acrescentando ao protótipo um componente chamado *gimbal*, que além de manter a câmera estabilizada, permite que a mesma gire. O *firmware* Copter já possui funcionalidades implementadas para o controle de *gimbal*, que permitem rotacionar a câmera em três eixos, podendo focar em um ponto de interesse, enquanto o *drone* sobrevoa a área.

Ao final do desenvolvimento do protótipo do *drone*, do software e dos testes realizados, pode-se concluir que a solução proposta é viável, possibilitando agilidade e praticidade na verificação de alarmes disparados. Sendo possível afirmar que o presente trabalho cumpriu os objetivos definidos no início da proposta, fornecendo um referencial teórico e prático para futuros trabalhos.

REFERÊNCIAS

ALMEIDA, Gian Lucas Nunes de. **Geração automática de código para a trajetória de VANTs em voos assistidos**. 2015. 75 f. TCC (Graduação) - Curso de Engenharia Elétrica, Centro de Ciências Tecnológicas, Universidade do Estado de Santa Catarina, Joinville, 2015. Disponível em: <<http://sistemabu.udesc.br/pergamumweb/vinculos/00000d/00000d34.pdf>>. Acesso em: 24 abr. 2018.

ANAC. **Regras da ANAC para uso de *drones* entram em vigor**. 2017. Disponível em: <http://www.anac.gov.br/noticias/2017/regras-da-anac-para-uso-de-drones-entram-em-vigor/release_drones_v2.pdf>. Acesso em 28 mar. 2018.

ANDRADE, Raúl Duarte Nóbrega. **Criação de uma estação de testes para um micro-helicóptero *autônomo***. 2012. 112 f. Dissertação (Mestrado) - Curso de Engenharia Informática, Universidade da Madeira, Funchal, 2012. Disponível em: <<http://hdl.handle.net/10400.13/595>>. Acesso em: 12 mar. 2018.

ANDRADE, Rodrigo de Oliveira. **Drones sobre o campo**. Revista Pesquisa Fapesp, São Paulo, n. 239, p. 74-77, jan. 2016. Disponível em: <https://issuu.com/pesquisafapesp/docs/pdf_239>. Acesso em: 12 mar. 2018.

BARATO, Bruno. **Projeto de um sistema de controle para veículos aéreos não tripulados**. 2014. 67 f. TCC (Graduação) - Curso de Engenharia Elétrica, Universidade de São Paulo, São Carlos, 2014. Disponível em: <<http://www.tcc.sc.usp.br/tce/disponiveis/18/180450/tce-26082014-152719>>. Acesso em: 18 mar. 2018.

BRASIL. AIC-N 17/18 – **Aeronaves remotamente pilotadas para uso recreativos aeromodelos**. Circular de Informações Aeronáuticas, DECEA, 2018. Disponível em: <<https://publicacoes.decea.gov.br/?i=publicacao&id=4782>>. Acesso em: 26 abr. 2018.

BRASIL. RBAC -E 94 – **Requisitos gerais para aeronaves não tripuladas de uso civil**. Publicado no DOU nº43, S/1, p.52, de 3 mai. 2017. Regulamento Brasileiro de Aviação Civil Especial, ANAC, 2017. Disponível em: <<http://www.anac.gov.br/assuntos/legislacao/legislacao-1/rbha-e-rbac/rbac/rbac-e-94-emd-00>>. Acesso em: 21 mar. 2018.

BREGU, Endri. **Re-engineering of aerial drone autopilot firmware with reactive programming**. 2015. 99 f. Dissertação (Mestrado) - Curso de Engenharia Informática, Eletrônica, Informação e Bioengenharia, Politecnico di Milano, Milão, 2015. Disponível em: <<http://hdl.handle.net/10589/108876>>. Acesso em: 28 fev. 2018.

CARVALHO, Nathalia Lacerda de; NETTO, Ana Luiza Coelho. **A gestão de áreas de risco a deslizamentos e estratégias de resposta a desastres: um estudo sobre a percepção de risco da população local através do sistema de alerta e alarme**. VI Congreso Iberoamericano de Estudios Territoriales y Ambientales, São Paulo, v. 123, n. 123, p.979-1000, 12 set. 2014. Disponível em: <<http://6cieta.org/arquivos-anais/eixo4/Nathalia%20Lacerda%20de%20Carvalho,%20Ana%20Luiza%20Coelho%20Netto.pdf>>. Acesso em: 1 jun. 2018.

CHIMENES, Rafael Martins. **Estudo e desenvolvimento de um quadrotor e sua estação de pilotagem remota**. 2017. 91 f. TCC (Graduação) - Curso de Ciência da Computação, Universidade Federal do Mato Grosso do Sul, Ponta Porã, 2017.

CHOUDHARI, Renuka Ramkishan; SANKESHWARI, Subhash.s.. **Modeling And Pid Cascade Control For Uav Type Quadrotor**. IOSR Journal Of Dental And Medical Sciences. Vol. 15, p. 52-58. ago. 2016. Disponível em: <<https://doi.org/10.9790/0853-1508095258>>. Acesso em: 24 mar. 2018.

CORRÊA, Mário Aparecido. **Modelo de veículos aéreos não tripulados baseado em sistemas multi-agentes**. 2008. 90 f. Tese (Doutorado) - Curso de Doutorado em Engenharia Elétrica, Departamento de Engenharia de Computação e Sistemas Digitais, Escola Politécnica da Universidade de São Paulo, São Paulo, 2008. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/3/3141/tde-28112016-111615>>. Acesso em: 18 mar. 2018.

DEMOLINARI, Humberto Cascardo. **Projeto de construção de um drone hexacóptero**. 2016. 58 f. TCC (Graduação) - Curso de Engenharia Mecânica, Engenharia Mecânica, Universidade Federal Fluminense, Niterói, 2016. Disponível em: <<https://app.uff.br/riuff/handle/1/1792>>. Acesso em: 29 mar. 2018.

FOUR, Ronald de et al. **Operational characteristics of brushless DC motors**. 123. ed. Kingston: 2007. 15 p. Disponível em: <http://www.eng.uwi.tt/depts/elec/staff/rdefour/Publications/Operational%20Characteristic%20of%20BLDCM_OA_WebPg.pdf>. Acesso em: 7 mar. 2018.

FOWERS, Spencer G. **Stabilization and Control of a Quad-Rotor Micro-UAV Using Vision Sensors**. 2008. 112 f. Tese (Mestrado) - Electrical And Computer Engineering, Ira A. Fulton College Of Engineering And Technology. Disponível em: <<http://hdl.lib.byu.edu/1877/etd2375>>. Acesso em: 28 fev. 2018.

FULLER, B; et al. **Hardware design and implementation of a MAVLink interface for an FPGA-based autonomous UAV flight control system**. Australia, Australia/Oceania, 2014. Disponível em: <<https://eprints.qut.edu.au/79541>>. Acesso em: 19 mar. 2018.

GABRIEL, Thiago Alberto Ramos. **Desenvolvimento de sistema de navegação de baixo custo de veículo terrestre não tripulado**. 2014. 86 f. Dissertação (Mestrado) - Curso de

Mestrado em Engenharia Mecânica, Departamento de Ciência e Tecnologia, Instituto Militar de Engenharia, Rio de Janeiro, 2014. Disponível em: <<http://www.ime.eb.br/arquivos/teses/se4/mec2014/ThiagoGabriel.pdf>>. Acesso em: 19 mar. 2018.

GIL, Antonio Carlos. **Como elaborar projetos de pesquisa**. 5. ed. São Paulo: Atlas, 2010. 184 p.

HOMA, Jorge M.. **Aeronaves e motores: Conhecimentos Técnicos**. 33. ed. São Paulo: Asa, 2013.

LAIRD, Jeremy (Ed.). **The Complete Guide to Drones**. [S. l.]: Future Publishing Limited, 2016. 148 p.

LUTZ, Mark; ASCHER, David. **Aprendendo Python**. 2. ed. Porto Alegre: Bookman, 2007. 568 p.

MARCONI, Marina de Andrade; LAKATOS, Eva Maria. **Fundamentos de metodologia científica**. 5. ed. São Paulo: Atlas, 2003. 305 p.

MARTY, Joseph A.. **Vulnerability Analysis of the MAVLink Protocol for Command and Control of Unmanned Aircraft**. 2014. 142 f. Tese (Doutorado) - Curso de Master Of Science In Cyber Operations, Department Of Electrical Computer Engineering, Air Force Institute Of Technology, Ohio, 2014. Disponível em: <<http://www.dtic.mil/docs/citations/ADA598977>>. Acesso em: 23 abr. 2018.

MEIER, Lorenz et al. **Pixhawk: A micro aerial vehicle design for autonomous flight using onboard computer vision**. *Autonomous Robots*, [s.l.], v. 33, n. 1-2, p.21-39, 23 fev. 2012. Springer Nature. Disponível em: <<http://dx.doi.org/10.1007/s10514-012-9281-4>>. Acesso em: 18 mar. 2018.

MELO, Alexandre Secchin de. **Implementação de um quadrotor como plataforma de desenvolvimento para algoritmos de controle**. 2010. 110 f. Dissertação (Mestrado) - Curso de Engenharia Elétrica, Universidade Federal do Espírito Santo, Vitória, 2010. Disponível em: <<http://repositorio.ufes.br/handle/10/6215>>. Acesso em: 17 mar. 2018.

MONICO, João Francisco Galera. **Posicionamento pelo GNSS: Descrição, fundamentos e aplicações**. 2. ed. São Paulo: Unesp, 2008. 476 p.

NETO, Milton Miranda. **Sistema Automático de Planejamento de Voos e Tratamento de Imagens para Veículos Aéreos Não Tripulados**. 2016. 86 f. Dissertação (Mestrado) - Curso de Pós-graduação em Engenharia Elétrica, Universidade Federal de Uberlândia, Uberlândia, 2016. Disponível em: <<https://repositorio.ufu.br/handle/123456789/14622>>. Acesso em: 17 abr. 2018.

OLIVEIRA, Edmar José Lopes de; SILVA, Flávio Rogério da; BARBOSA, Luis Filipe Wiltgen. **Dispositivo inercial baseado em giroscópio para aplicação em Robótica**. Jacareí: XXI Encontro Latino Americano de Iniciação Científica, XVII Encontro Latino Americano de Pós-graduação, XI INIC JR, 2008. Disponível em:

<http://www.inicepg.univap.br/cd/INIC_2008/anais/arquivosINIC/INIC0980_01_A.pdf>. Acesso em: 18 mar. 2018.

PATELLA, Luciana. **Veículos Aéreos Não Tripulados**: os novos aliados da Engenharia. Revista Bimestral do Conselho Regional de Engenharia e Agronomia do Rio Grande do Sul, Rio Grande do Sul, n. 116, p. 28-31, out. 2016. Disponível em: <<http://www.crea-rs.org.br/site/arquivo/revistas/Ed116.pdf>>. Acesso em: 14 mar. 2018.

PECHARROMÁN, José María Peral; VEIGA, Ricardo. **Estudo Sobre a Indústria Brasileira e Europeia de Veículos Aéreos Não Tripulados**. Brasília: Diálogos Setoriais União Europeia - Brasil, 2017. 92 p. Disponível em: <http://www.mdic.gov.br/images/publicacao_DRONES-20161130-20012017-web.pdf>. Acesso em: 25 mar. 2017.

ROCHA, José Antônio M. R. **GPS: Uma abordagem prática**. 4. ed. Recife: Bagaço, 2003. 232 p.

SANTOS, Thobias Rodolfo S. dos. **Integração de um VANT**. 2015. 41 f. TCC (Graduação) - Curso de Engenharia Elétrica/eletrônica, Universidade do Vale do Paraíba, São José dos Campos, 2015. Disponível em: <<http://biblioteca.univap.br/dados/000015/00001534.pdf>>. Acesso em: 18 mar. 2018.

SCUSSEL, Alexandre. **As mil e uma utilidades dos drones**. Revista Mundo GEO, Paraná, n. 83, p. 20-24, set. 2015. Disponível em: <https://issuu.com/editoramundogeo/docs/mundogeo_83>. Acesso em: 12 mar. 2018.

SHARMA, Sajal et al. **UAV for Surveillance and Environmental Monitoring**. Indian Journal Of Science And Technology. Uttarakhand, p. 1-4. nov. 2016. Disponível em: <<https://doi.org/10.17485/ijst/2016/v9i43/104396>>. Acesso em: 18 abr. 2018.

SILVA, Alberto Rogério e. **HEALTHDRONES - Uma Plataforma em Software para Controle de Veículos Não Tripulados e Planejamento de Trajetórias**. 2016. 92 f. Dissertação (Mestrado) - Curso de Pós-graduação em Informática Aplicada, Departamento de Estatística e Informática, Universidade Federal Rural de Pernambuco, Recife, 2016. Disponível em: <http://www.ppgia.ufrpe.br/sites/ww4.ppgia.ufrpe.br/files/documentos/2016_-_healthdrones_-_uma_plataforma_em_software_para_controle_de_veiculos_ao_tripulados_e_planejamento_de_trajetorias.pdf>. Acesso em 18 mar. 2018.

SILVA, Fernando Ginez da. **Reconhecimento de movimentos humanos utilizando um acelerômetro e inteligência computacional**. 2013. 90 f. Dissertação (Mestrado) - Curso de Engenharia, Departamento de Engenharia de Sistemas Eletrônicos, Escola Politécnica da Universidade de São Paulo, São Paulo, 2013. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/3/3140/tde-22092014-103901>>. Acesso em: 18 mar. 2018.

SUN, Yue. **Modeling, identification and control of a quad-rotor drone using low-resolution sensing**. 2012. 70 f. Tese (Doutorado) - Curso de Engenharia Mecânica, Mechanical Sci &

Engineering, University Of Illinois At Urbana-champaign, Urbana, 2012. Disponível em: <<http://hdl.handle.net/2142/34205>>. Acesso em: 09 abr. 2018.

UBAYA, Huda; IQBAL, Muhammad. *First Person View on Flying Robot For Real Time Monitoring*. ICON-CSE, Palembang, v. 1, n. 1, p.41-43, out. 2014. Disponível em: <<http://seminar.ilkom.unsri.ac.id/index.php/iconcse/article/view/14>>. Acesso em: 12 mai. 2018.

VILLATE, Jaime E.. **Dinâmica e Sistemas Dinâmicos**. [Porto]: Jaime E. Villate, 2013. 283 p. E-book. Disponível em: <http://www.villate.org/doc/fisica1/dinamica_20130320.pdf>. Acesso em: 10 maio 2018.

YADAV, Shubham; SHARMA, Monika; BORAD, Abhishek. *Thrust Efficiency of Drones (Quad Copter) with Different Propellers and Their Payload Capacity*. International Journal Of Aerospace And Mechanical Engineering. [s. L.], p. 18-23. 2 abr. 2017. Disponível em: <<http://www.ijamejournals.com/pdf/rpm11174.pdf>>. Acesso em: 01 abr. 2018.

ZUQUETTE, Lazaro. **Riscos, desastres e eventos naturais perigosos: Aspectos conceituais na análise e estimativa de riscos**. [s. L.]: Elsevier Brasil, 2017. 304 p.