

MINISTÉRIO DA DEFESA  
EXÉRCITO BRASILEIRO  
DEPARTAMENTO DE CIÊNCIA E TECNOLOGIA  
INSTITUTO MILITAR DE ENGENHARIA  
CURSO DE MESTRADO EM SISTEMAS E COMPUTAÇÃO

IGOR DEVULSKY PRATA

METODOLOGIA PARA LOCALIZAÇÃO DE ATIVOS POR  
ALGORITMOS DE FUSÃO DE SENSORES PARA  
RASTREAMENTO AUTÔNOMO

Rio de Janeiro  
2018

**INSTITUTO MILITAR DE ENGENHARIA**

**IGOR DEVULSKY PRATA**

**METODOLOGIA PARA LOCALIZAÇÃO DE ATIVOS POR  
ALGORITMOS DE FUSÃO DE SENSORES PARA  
RASTREAMENTO AUTÔNOMO**

Dissertação de Mestrado apresentada ao Curso de Mestrado em Sistemas e Computação do Instituto Militar de Engenharia, como requisito parcial para a obtenção do título de Mestre em Ciências em Sistemas e Computação.

Orientador: TC Anderson Fernandes Pereira dos Santos - D.Sc.  
Co-Orientador: Prof. Paulo Fernando Ferreira Rosa - Ph.D.

Rio de Janeiro  
2018

c2018

INSTITUTO MILITAR DE ENGENHARIA  
Praça General Tibúrcio, 80 - Praia Vermelha  
Rio de Janeiro - RJ CEP 22290-270

Este exemplar é de propriedade do Instituto Militar de Engenharia, que poderá incluí-lo em base de dados, armazenar em computador, microfilmar ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade do(s) autor(es) e do(s) orientador(es).

004.69 Prata, Igor Devulsky  
S586e Metodologia para Localização de Ativos por Algoritmos de Fusão de Sensores para Rastreamento Autônomo / Igor Devulsky Prata, orientado por Anderson Fernandes Pereira dos Santos e Paulo Fernando Ferreira Rosa - Rio de Janeiro: Instituto Militar de Engenharia, 2018.

172p.: il.

Dissertação (mestrado) - Instituto Militar de Engenharia, Rio de Janeiro, 2018.

1. Curso de Sistemas e Computação - teses e dissertações. 1. Wardriving. 2. Localização. 3. VANT. 4. Drone. 5. Fusão de Sensores. 6. radiofrequência. I. dos Santos, Anderson Fernandes Pereira . II. Rosa, Paulo Fernando Ferreira . III. Título. IV. Instituto Militar de Engenharia.

**INSTITUTO MILITAR DE ENGENHARIA**

**IGOR DEVULSKY PRATA**

**METODOLOGIA PARA LOCALIZAÇÃO DE ATIVOS POR  
ALGORITMOS DE FUSÃO DE SENSORES PARA  
RASTREAMENTO AUTÔNOMO**

Dissertação de Mestrado apresentada ao Curso de Mestrado em Sistemas e Computação do Instituto Militar de Engenharia, como requisito parcial para a obtenção do título de Mestre em Ciências em Sistemas e Computação.

Orientador: TC Anderson Fernandes Pereira dos Santos - D.Sc.

Co-Orientador: Prof. Paulo Fernando Ferreira Rosa - Ph.D.

Aprovada em 02 de Fevereiro de 2018 pela seguinte Banca Examinadora:

---

TC Anderson Fernandes Pereira dos Santos - D.Sc. do IME - Presidente

---

Prof. Paulo Fernando Ferreira Rosa - Ph.D. do IME

---

TC Antônio Eduardo Carrilho da Cunha - D.Sc. do IME

---

Prof. Fábio Borges de Oliveira - Dr.-Ing. do LNCC

Rio de Janeiro  
2018

Esse trabalho é dedicado a todos que acreditam na ciência como investimento para o futuro de um país.

## AGRADECIMENTOS

Agradeço aos meus orientadores, Anderson Santos e Paulo Rosa que tornaram possível a execução de um projeto multidisciplinar de grande complexidade em um curto espaço de tempo. Esse trabalho não seria possível sem a colaboração de diversos pesquisadores, familiares e amigos que o apoiaram em diversas fases: à minha esposa, Larissa Prata, pelo constante suporte e pelo fornecimento de equipamentos; aos doutorandos Luís Silva e Ricardo Maróquio, por me ensinarem a construir um VANT totalmente customizável; aos graduandos Elmo Brandão, Gustavo Couto, Rodrigo Lopes e Yugo Nihari por desenvolverem peças e ferramentas indispensáveis ao ensaio; aos veteranos Andressa Siqueira e Fábio Costa por compartilharem suas experiências na mesma linha de pesquisa. Em especial à toda minha família e amigos pela compreensão e apoio nesse momento de dedicação tão intensa de minha vida.

“A vida é a arte do encontro.”

VINÍCIUS DE MORAES

## SUMÁRIO

LISTA DE ILUSTRAÇÕES .....	11
LISTA DE TABELAS .....	15
LISTA DE ALGORITMOS .....	16
LISTA DE SIGLAS .....	17
LISTA DE ABREVIATURAS .....	19
<b>1 INTRODUÇÃO .....</b>	<b>22</b>
1.1 Motivação .....	23
1.1.1 Alinhamento com a Estratégia Nacional de Defesa .....	24
1.2 Caracterização do Problema .....	25
1.3 Objetivo .....	28
1.4 Contribuições Esperadas .....	28
1.5 Organização da Dissertação .....	29
<b>2 TÓPICOS TUTORIAIS .....</b>	<b>30</b>
2.1 Radiofrequência .....	30
2.2 Wireless Sensor Network .....	31
2.2.1 GPS (e outros sistemas de localização por satélite) .....	31
2.2.2 Redes GSM .....	32
2.2.3 Características do padrão IEEE 802.11 .....	33
2.2.4 Características do padrão Bluetooth .....	34
2.3 Dispositivos de Localização de Emergência .....	35
2.4 Localização <i>Indoor</i> e <i>Outdoor</i> .....	36
2.5 Localização por <i>Wardriving</i> .....	37
2.5.1 Principais Ferramentas para <i>Wardriving</i> .....	38
2.6 Fusão de Sensores .....	39
2.7 Localização por Radiofrequência .....	41
2.8 Veículos Aéreos não Tripulados .....	45
2.9 Trabalhos Relacionados .....	47

<b>3</b>	<b>METODOLOGIA</b>	52
3.1	Determinação da técnica de localização	53
3.2	Arquitetura proposta	54
3.2.1	<i>Framework</i> Metodológico	55
3.2.2	Estimativa de Distância	57
3.2.3	Estimativa de Posição	59
3.2.4	Fusão de Sensores	61
3.3	Veículo Aéreo não Tripulado	62
3.3.1	Sistema Embarcado	63
3.3.1.1	Computador de Bordo	63
3.3.1.2	Unidade de Medidas Inerciais (IMU)	64
3.3.1.3	Sensores	65
3.3.1.4	Adaptações	66
3.3.2	Sistema de Navegação	66
3.3.3	Estação de Controle em Solo	68
3.4	Desenvolvimento em <i>Software</i>	69
<b>4</b>	<b>ENSAIOS</b>	75
4.1	Ensaio iniciais	76
4.2	Ensaio em ambientes controlados	78
4.3	Ambiente de simulação de missões de voo	78
4.4	Ensaio de coleta de Sinal	81
4.4.1	Testes pelo computador de bordo	81
4.4.2	Testes com o VANT sem hélices	83
4.4.3	Testes com o VANT em voo controlado	85
4.5	Testes em missões de voo autônomo	86
4.6	Conclusões da fase experimental	89
<b>5</b>	<b>CONSIDERAÇÕES FINAIS</b>	93
5.1	Conclusões	93
5.2	Trabalhos Futuros	95
5.3	Agradecimentos Institucionais	96
<b>6</b>	<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>	97

<b>7</b>	<b>APÊNDICES</b>	105
7.1	APÊNDICE 1: Lista de Componentes para Montagem do Quadricóptero	106
7.2	APÊNDICE 2: Informações do voo experimental	108
7.2.1	Massa de dados	108
<b>8</b>	<b>ANEXOS</b>	113
8.1	ANEXO 1: <i>Check-list</i> de Equipamentos	114
8.1.1	Lista de ferramentas necessárias para testes de voos com <i>drones</i> :	115
8.1.2	Energia	115
8.1.3	Ferramentas	115
8.1.4	Informática	116
8.1.5	Registro	116
8.1.6	Apoio	117
8.2	ANEXO 2: Procedimento Operacional	118
8.2.1	Lista de passos operacionais para missões com VANT	119
8.2.2	Preparativos antes do Ensaio	119
8.2.3	Preparativos em Campo	120
8.2.4	Atitude em caso de Incidente	123
8.3	ANEXO 3: Guia de montagem do VANT	124
8.3.1	Inventário	125
8.3.2	Equipamentos principais	125
8.3.3	Modelagem de Peças 3D	125
8.3.4	Itens Extras e ferramentas	127
8.3.5	Passos de montagem	128
8.3.6	Apresentação dos Componentes	128
8.3.7	Passos para Construção	129
8.3.8	Montagem do Sonar	139
8.3.9	Conclusões sobre o processo de montagem	139
8.4	ANEXO 4: Especificação de <i>Hardware</i> Embarcado	141
8.4.1	Rádios	142
8.4.1.1	WiFi	142
8.4.1.2	Bluetooth	143
8.4.1.3	Comando e Telemetria	144
8.4.2	Alimentação	145
8.4.3	Sensores de bordo	145

8.4.4	Dispositivo Alvo .....	146
8.5	ANEXO 5: Procedimento de Instalação e Configuração do Simulador de Voo	148
8.5.1	Instalação do ambiente simulado.....	149
8.5.2	Instalação .....	149
8.5.3	Configuração inicial do ambiente simulado .....	149
8.5.4	Execução do ambiente de simulação .....	151
8.5.5	Compilando SITL a partir do seu código fonte .....	153
8.5.5.1	Baixando o Ardupilot .....	153
8.5.5.2	Instalando pacotes necessários .....	153
8.5.5.3	Configurando o ambiente .....	154
8.5.5.4	Iniciando o SITL simulator .....	154
8.6	ANEXO 6: Procedimento de instalação e configuração do sistema de bordo .	155
8.6.1	Instalando o Sistema Operacional .....	156
8.6.2	Configurando o acesso WiFi .....	156
8.6.3	Testando as configurações de Rede .....	157
8.6.4	Problemas Conhecidos .....	158
8.6.5	Configurando o S.O. e os principais serviços .....	158
8.6.6	Configurando os Serviços essenciais .....	159
8.6.6.1	<i>Setup</i> mandatório do Ardupilot .....	161
8.6.6.2	Configuração de Enxames ( <i>Swarm</i> ) .....	161
8.6.6.3	Configuração do Sonar Maxbotix I2C .....	161
8.6.6.4	Configuração do Range Finder (Garmin Lidar Lite V3) .....	162
8.6.7	Configuração de Gimbal.....	164
8.6.8	Revisar configurações de Rotores .....	164
8.6.9	Dependências de software .....	164
8.6.10	Ferramentas Bluetooth.....	165
8.6.11	Ferramentas WiFi .....	166
8.6.12	Ferramentas ZWave .....	166
8.6.13	Configurar placas de Redes .....	166
8.6.14	TP-Link T2UH (Ralink mt7610u) .....	166
8.6.15	TP-Link T4UH (AC1200) .....	167
8.6.16	Comandos úteis .....	168
8.6.17	Comandos Bluetooth .....	168
8.6.18	Comandos Bluetooth BLE.....	170
8.6.19	Comandos WiFi .....	171

## LISTA DE ILUSTRAÇÕES

FIG.1.1	Mapa do estado do Rio de Janeiro com a distribuição de antenas celulares GSM, os pontos em roxo se referem a amostras com alta recepção e em marrom, em baixa (Fonte: OpenSignal) . . . . .	26
FIG.1.2	Uma rota de voo autônoma de VANT com regiões de sinais disponíveis . . . . .	27
FIG.2.1	Mapa de São Francisco (EUA) com todos os <i>Access Points</i> (APs) reconhecidos pelo Projeto WiGLE (Wireless Geographic Logging Engine) em 11/2016. Os pontos em vermelho demonstram a localização de APs com alta precisão (devido ao número de amostras obtidas) e em verde a posição de APs com baixa precisão. Em azul são considerados APs ainda sem confirmação de localização, ou errantes. (licença CC BY-SA) . . . . .	38
FIG.2.2	Compartimentação do problema de localização em WSN. . . . .	42
FIG.2.3	Pontos de coleta insuficiente para trilateração . . . . .	44
FIG.2.4	Ponto exato de estimativa de posição por trilateração . . . . .	45
FIG.2.5	VANT de asa rotativa do tipo quadrirrotor utilizado nesse trabalho . . . .	46
FIG.3.1	Taxonomia de técnicas de localização em WSN. . . . .	52
FIG.3.2	<i>Framework</i> proposto para a metodologia de localização . . . . .	56
FIG.3.3	Tolerâncias a pontos de estimativa fora da interseção dos círculos raio $d_i$ . . . . .	60
FIG.3.4	VANT utilizado nos testes de voo, com carga paga de até 1,2 kg. . . . .	62
FIG.3.5	Placa Raspberry Pi 3 Modelo B com o detalhamento de todas as suas interfaces . . . . .	64
FIG.3.6	Tela de edição de missões de voo pelo GCS QGroundControl . . . . .	68
FIG.3.7	Exemplo de saída em mapa do Google dos resultados de um voo . . . . .	71
FIG.3.8	Relação entre componentes de <i>software</i> . . . . .	72
FIG.3.9	Fluxo de controle da aeronave por <i>software</i> . . . . .	73
FIG.3.10	Protótipo do sistema computacional embarcado com os sensores. . . . .	74
FIG.4.1	Esquema para medições reais entre um VANT e um transmissor de RF . . . . .	75

FIG.4.2	Posições de medições de distância entre um VANT e um transmissor de RF e o momento da estimativa de localização .....	76
FIG.4.3	Ensaio de coleta apenas com o computador de bordo .....	77
FIG.4.4	Fluxo de controle dos componentes envolvidos na solução de <i>Software in the loop</i> .....	79
FIG.4.5	GCS com aeronave EMULADA .....	80
FIG.4.6	PROTÓTIPO em execução local durante simulação de voo .....	81
FIG.4.7	Ensaio de coleta com o VANT em ambiente controlado e sem voo .....	82
FIG.4.8	Coletas de sinais WiFi e BLE com o VANT a 1m de distância do emissor e diferentes fontes de alimentação e antenas .....	83
FIG.4.9	Ensaio de coleta com o VANT em ambiente controlado com os rotores ativados .....	84
FIG.4.10	Coletas de sinais WiFi com o VANT a 1,72m de distância do emissor e diferentes potências de <i>throttle</i> .....	85
FIG.4.11	Ensaio de coleta com o VANT em voo controlado .....	86
FIG.4.12	Coletas de sinais WiFi com o VANT em voo controlado a 1,72m de distância do emissor .....	86
FIG.4.13	Pontos da missão de voo de 17 de novembro de 2017 em Cachoeiro de Itapemirim no ES .....	89
FIG.4.14	Pontos de coleta obtidos durante o ensaio de voo de 17 de novembro de 2017 em Cachoeiro de Itapemirim no ES, com as respectivas distâncias calculadas e estimativas de localização ao centro .....	90
FIG.4.15	Sensores de altitude da aeronave durante o ensaio de voo de 17/11/2017	91
FIG.4.16	Coleta de valores de RSSI durante o ensaio de voo de 17/11/2017 .....	92
FIG.4.17	Voo experimental sendo realizado em região agrícola no interior do Espírito Santo .....	92
FIG.7.1	Log do voo de 17 de novembro de 2017 em Cachoeiro de Itapemirim no ES .....	109
FIG.8.1	Exemplo de componente 3D presente no arquivo de peças para impressão: Casco Inferior do sistema computacional embarcado .....	126
FIG.8.2	Itens da estrutura principal do <i>frame</i> DJI modelo F450 .....	128
FIG.8.3	Kit de propulsores, hélice e ESCs DJI modelo 4205 .....	129
FIG.8.4	Protetores de Hélices para o <i>frame</i> DJI F450 .....	129

FIG.8.5	Módulo de câmera Raspberry Pi v.2 e cabos <i>flat</i> de tamanhos diferentes	130
FIG.8.6	Placa de piloto automático NAVIO2 da Emlid	130
FIG.8.7	Sonar Maxbotix modelo EZ0 Digital	131
FIG.8.8	Casco customizado para montagem do sistema computacional embarcado. Impressão em plástico PLA	131
FIG.8.9	Kit de rádios para telemetria 3DR 433MHz	132
FIG.8.10	Processo de soldagem da placa de distribuição elétrica do <i>frame</i> F450, ESCs e conector de alimentação da bateria	132
FIG.8.11	Placa de distribuição com todos os terminais já soldados e isolados com o auxílio de pasta isolante elétrica	133
FIG.8.12	Processo de montagem dos raios do <i>frame</i> F450, fixando à placa distribuidora de energia	133
FIG.8.13	Fixação do ESC com braçadeiras de nylon	134
FIG.8.14	Fixação dos rotores por parafusos e passagem dos cabos dos rotores pela estrutura do <i>frame</i>	134
FIG.8.15	Acabamento com fita isolante líquida nos pontos de contato ainda expostos	135
FIG.8.16	Acabamento sendo aplicado em peças em plástico que contém imperfeições. Uma broca de precisão foi utilizada em algumas partes para nivelar os furos para o encaixe da placa Raspberry Pi	135
FIG.8.17	Montagem final dos cascos e estabilizadores em borracha	136
FIG.8.18	Ordem de montagem das placas que constituem o computador de bordo do VANT (Fonte: Emlid)	136
FIG.8.19	Montagem final do casco de proteção do computador de bordo da aeronave	137
FIG.8.20	Diagrama da placa Navio2 com destaque para a conexão com o modulo de potência (Fonte: Emlid)	137
FIG.8.21	Parte traseira do computador de bordo com os terminais PPM e PWM exposto para conexão com o receptor de rádio e ESCs dos motores	138
FIG.8.22	Ordem de pinagem dos conectores PWM na Navio2 (Fonte: Emlid)	138
FIG.8.23	Diagrama de conexão de periféricos na placa Navio2	139
FIG.8.24	Fixação do sonar Maxbotix EZ0 no case, impresso em plástico PLA, na base do <i>frame</i> F450 - foram utilizadas borrachas de amorteci-	

mento .....	140
FIG.8.25 Cabeamento I2C na placa Navio2. A ordem dos cabos deve ser seguida ao soldá-lo no sonar .....	140
FIG.8.26 GCS com aeronave EMULADA .....	150
FIG.8.27 Tela inicial do comando Rasp-Config .....	159
FIG.8.28 Tela de configuração de <i>Range Finder</i> no GCS APM Planner2 .....	162
FIG.8.29 Tela de configuração de Gimbal no GCS APM Planner2 .....	164
FIG.8.30 Sentido de rotação dos rotores (Fonte: Emlid) .....	165

## LISTA DE TABELAS

TAB.2.1	Diferenças entre as especificações comerciais do padrão IEEE 802.11 33	
TAB.2.2	Especificações de largura de banda e distância de transmissão de cada versão do padrão bluetooth (Fonte: SIG (2016))	34
TAB.2.3	Comparação de largura de banda e consumo médio de energia de diferentes estados entre dispositivos GSM, WiFi e bluetooth	35
TAB.2.4	Classificação de VANTs pelo tipo de asa	46
TAB.2.5	Comparação entre Trabalhos Relacionados	51
TAB.3.1	Comparação entre técnicas de estimativa de distância em WSN	54
TAB.3.2	Radiotransmissores embarcados no VANT	65
TAB.4.1	Tabela com o cálculo de distâncias efetuado a partir de amostras a 1m de distância do alvo	84
TAB.4.2	Tabela com o tempo de execução de coletas por número de amostras em missão de voo similar a referenciada pela tabela 7.5	87
TAB.7.1	Lista de Componentes, com seus fornecedores e preços, para a mon- tagem do quadrirrotor utilizado na pesquisa Fonte: (RICARDO M. BERNARDO; ROSA, 2017)	107
TAB.7.2	Total de radiotransmissores IEEE 802.11 detectados	108
TAB.7.3	Massa de dados coletados no voo de 17/11/2017	110
TAB.7.4	Massa de dados coletados no voo de 17/11/2017	111
TAB.7.5	Massa de pontos ( <i>waypoints</i> ) em missão autônoma no voo de 17/11/2017 112	

## LISTA DE ALGORITMOS

1	Ordem de execução do algoritmo de localização . . . . .	57
2	Execução do algoritmo de estimativa de distância . . . . .	59
3	Execução do algoritmo de estimativa de posição . . . . .	61

## LISTA DE SIGLAS

ADC	Analog-to-Digital Converter
ANAC	Agência Nacional de Aviação Civil
ANATEL	Agência Nacional de Telecomunicações
AOA	Angle of Arrival
AP	Access Point
API	Application Programming Interface
ARM	Acorn RISC Machines
ARP	Aeronave Remotamente Pilotada
BLE	Bluetooth Low Energy
CPU	Central Processing Unit
CSV	Comma-Separated Values
DECEA	Departamento de Controle do Espaço Aéreo
END	Estratégia Nacional de Defesa
EPIRB	Emergency position-indicating radio beacons
ERB	Estação Rádio Base
ESC	Electronic Speed Control
FPS	Frames por Segundo
GCS	Ground Control System
GLONASS	GLOBAL'naya NAVigatsionnaya Sputnikovaya Sistema
GNSS	Global Navigation Satellite System
GNU	GNU is not a Unix
GPIO	General Purpose Input/Output
GPS	Global Positioning System
GSM	Global System for Mobile
HTML	Hyper Text Markup Language
I2C	Inter-Integrated Circuit
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos
IME	Instituto Militar de Engenharia
IMU	Inertial Measurement Unit
IoT	Internet of Things
JSON	JavaScript Object Notation
Lidar	Light Detection And Ranging
MP	Mega Pixel

NoAA	National Oceanic and Atmospheric Administration
PCB	Printed Circuit Board
PLB	Personal Locator Beacon
PPM	Pulse Position Modulation
PWM	Pulse Width Modulation
RC	Remote Control
RPA	Remotely Piloted Aircraft
RPAS	Remotely Piloted Aircraft System
RSSI	Received Signal Strength Indicator
SDK	Software Development Kit
SGDBD	Sistema de Gestão de Base de Dados
SITL	Software in the Loop
SLAMM	Selective Look-Ahead Map-Matching
SO	Sistema Operacional
SSH	Secure Shell
SSID	Service Set Identifier
SUAS	Small Unmanned Air/Aircraft System
TCP	Transmission Control Protocol
TDOA	Time Difference of Arrival
TOA	Time of Arrival
TTY	TeleTYpe
UART	Universal Asynchronous Receiver/Transmitter
UAS	Unmanned Air/Aircraft System
UAV	Unmanned Aerial Vehicle
UBEC	Ultimate Battery Eliminator Circuit
UDP	User Datagram Protocol
USB	Universal Serial Bus
VANT	Veículo Aéreo não Tripulado
VGA	Video Graphic Array
WPS	Wireless Positioning Systems
WSN	Wireless Sensor Network

## LISTA DE ABREVIATURAS E SÍMBOLOS

### ABREVIATURAS

C	-	Constante Ambiental
A0	-	Medida do RSSI médio recebido a uma distância constante
WF	-	WiFi
BT	-	Bluetooth

### SÍMBOLOS

$RSS$	-	Potência de sinal recebida pelo emissor
$P_0$	-	Potência recebida pelo sensor a uma distância de 1 metro
$\alpha$	-	Componente ambiental de path loss
$X$	-	Ponto de melhor aproximação da posição real P
$L_i$	-	distância entre cada ponto de coleta

## RESUMO

A crescente evolução tecnológica pode ser observada no avanço dos padrões de comunicação digital, expandindo os seus aspectos técnicos de transmissão e suas aplicações. A alta disponibilidade tecnológica, sua miniaturização e a facilidade de acesso possibilitaram diversos cenários de aplicação, principalmente na computação embarcada. Essa evolução é refletida na queda de preços de produção de bens de consumo, na popularidade de *smartphones* e computadores transportáveis - que contam com diversos recursos de comunicação embarcados. O acesso aos sistemas robóticos possibilita que veículos aéreos não tripulados (VANTs) de baixo custo possam efetuar tarefas com considerável nível de automação, propiciando maior diversidade de aplicações.

Apesar da popularidade de padrões específicos de localização por satélite (GNSS, *Global Navigation Satellite System*), sua aplicação é limitada por fatores ambientais, como dentro edificações, onde o sinal é consideravelmente enfraquecido ou obliterado por obstáculos. Dentro desse cenário, sistemas originalmente utilizados para comunicação desempenham um papel versátil, provendo recursos de localização aos dispositivos em sua proximidade que por sua vez podem estimar a sua distância do transmissor a partir do decaimento de sinal recebido.

Este trabalho propõe uma técnica que utiliza rádios em diferentes padrões com a finalidade de melhorar a estimativa de localização de uma fonte emissora de radiofrequência, através da aplicação de algoritmos de fusão de sensores. A estimativa de posicionamento por um único tipo de sensor gera imprecisões que podem ser acentuadas dependendo do meio. Diversos trabalhos sugerem que o problema de localização não deve ser tratado por uma abordagem sensorial única, que compreenda todos os desafios da área, mas sim pelo uso de múltiplas técnicas que possam sobrepor limitações e aumentar a acuidade do sistema.

Essa dissertação apresenta uma metodologia desenvolvida para ser embarcada em VANTs, executando aplicações de *wardriving* em protocolos de rede bem difundidos, WiFi (IEEE 802.11) e bluetooth, mas com o potencial de suportar outros padrões, possibilitando à aeronave detectar alvos emissores de maneira autônoma. A técnica adotada utilizou fusão de sensores para incorporar dados de RSSI (*Received Signal Strength Indicator*), recebidos por rádios WiFi e bluetooth, na estimativa de distância e trilateração por múltiplos pontos de análise para determinar a posição do alvo. A metodologia foi validada em diversos testes simulados e em ambientes controlados para medir a performance e aplicabilidade. Foram realizados testes de voo reais para detecção de único *smartphone*, que atuou como transmissor de sinais em múltiplos padrões.

Os resultados em campo aberto se mostraram precisos dentro dos limites operacionais de cada rádio adotado na fase experimental. Durante a confecção do trabalho foram gerados artefatos (presentes nos anexos dessa dissertação) sobre a implementação do *software* e montagem do sistema computacional embarcado, massa de dados coletados em voo, bem como detalhamentos sobre a construção da aeronave experimental e procedimentos operacionais utilizados nos ensaios.

## ABSTRACT

The increasing technological evolution can be observed in the advance of digital communication standards, expanding its technical specifications and applications. The high technological availability, its miniaturization, and access allowed several application scenarios, especially in embedded computing. This evolution is reflected in the prices of consumer goods, smartphones and other computer gadgets, which have several embedded communication capabilities. The access to robotic system enables low-cost unmanned aerial vehicles (UAVs) to perform tasks with a considerable automation, providing a greater diversity of applications.

Despite the popularity of satellite location standards (GNSS, Global Navigation Satellite System), its application is limited by environmental factors, such as indoor use, where the signal is considerably weakened or obliterated by obstacles. Within this scenario, systems originally used for communication play a versatile role: providing location resources to devices in their vicinity which, in turn, can estimate their distance from the transmitter from the received signal decay.

This work proposes a technique that uses radios in different standards in order to improve location estimation of a radiofrequency source through the application of sensor fusion algorithms. Positions estimated by a single type of sensor generate inaccuracies that can be accentuated depending on the environment. Several studies suggest that the localization problem should not be addressed by a single sensory approach, that fulfill all the challenges of the area, but rather by multiple techniques that can overcome limitations and increase the precision of the system.

This dissertation presents a methodology to be embedded in UAV, executing wardrive applications in well-known network protocols, WiFi (IEEE 802.11) and bluetooth, with the potential to support other standards, allowing the aircraft to autonomously detect emitting targets during a flight. The adopted technique uses sensor fusion to incorporate RSSI (Received Signal Strength Indicator) data, received by WiFi and bluetooth radios, to estimate distance and trilateration, by multiple points of analysis, to determine the position of a target. The methodology was validated in several simulated tests and in controlled environments to measure performance and applicability. Real flight tests were performed to detect a smartphone, which acted as a multi-pattern signal transmitter.

The results in the open field were accurate within the operational limits of each radio adopted during the experimental phase. This work generated auxiliary artifacts (available in the annexes of this dissertation) about the software implementation and assembly of the onboard computer system, a mass data collected in flight, and further details on the construction of the experimental aircraft and operational procedures used on the experiments.

# 1 INTRODUÇÃO

Programas e dispositivos sensíveis ao contexto são cada vez mais comuns e suas aplicações são inúmeras. Atualmente, o seu uso varia desde a localização de um usuário dentro do espaço, com suas coordenadas estimadas por diversos sensores, até a detecção de outros dispositivos no ambiente, determinando o seu posicionamento através de dados gerados em tempo real ou por meio de serviços dedicados ao gerenciamento de grandes massas de dados de localização (SAPIEZYNSKI et al., 2015).

Dentro do contexto atual, produtos eletrônicos de consumo como *smartphones*, possuem sensores embarcados como receptores de GPS (*global positioning system*), antenas WiFi (no padrão IEEE 802.11) e transmissores bluetooth. Esses recursos são cada vez mais difundidos em *hardwares* pessoais de uso diário. Padrões de radiotransmissores, destinados originalmente à comunicação de dados, também podem ser tratados como sensores de posicionamento (ANANTHANARAYANAN; STOICA, 2009). Técnicas acessíveis para mapeamento de pontos de acesso WiFi, como o *wardriving*, são aplicadas com o propósito de estimativa de distância *indoor* (HAEBERLEN et al., 2004) e *outdoor*, como uma alternativa viável aos sistemas de localização por satélite.

Dispositivos de localização emergencial, como os EPIRBs (*Emergency position-indicating radio beacons*) para uso náutico, existem há décadas. Apesar disso, a utilização desses aparelhos não é popularizada, seja pelo seu volume, especificidade de uso ou pelo alto custo (MUNSHI et al., 2014). A detecção de vítimas em ambientes de tragédia, como inundações, ou em locais onde uma rede de comunicação não está estruturada, como florestas, é um trabalho extremamente custoso para as equipes de resgate empenhadas, que muitas vezes só dispõem de observações não automatizadas para a localização.

Veículos aéreos não tripulados (VANTs) rádio controlados são tecnologias de crescente popularidade no mercado de consumo. Diversos fabricantes disponibilizam produtos desse tipo já prontos para voo (*ready to fly*), mas também há uma grande demanda por peças e equipamentos. Comunidades de usuários dedicadas à tecnologia desenvolvem plataformas, *softwares* e procedimentos *open source* para a criação e customização, impulsionando ainda mais o seu uso e fomentando projetos para diversos cenários.

Apesar da grande presença em meios urbanos e industriais e com altas bandas de transmissão, redes de dados sem fio ainda são heterogêneas, existindo hoje em uma ampla variedade de padrões para diferentes casos de uso. Padrões com grande adesão, como o

WiFi, são sensíveis a ruído e o seu sinal em alta frequência é facilmente obliterável por obstáculos. Porém, a disponibilidade por variados sistemas de comunicação *wireless* em um mesmo ambiente - em parte, graças ao avanço do IoT (Internet das Coisas), expande a gama de possibilidade de identificação de uma infraestrutura a partir de seus recursos tecnológicos transmissores de radiofrequência.

A presente dissertação de mestrado, apresenta uma alternativa de baixo custo para localização automatizada de alvos transmissores de sinal de radiofrequência em múltiplos padrões. Mais especificamente, foi desenvolvida uma aplicação prática em um veículo aéreo não tripulado para a detecção de radiotransmissores compatíveis com os padrões IEEE 802.11 e bluetooth, empregando fusão de sensores de modo a oferecer maior redundância e taxa de acerto em suas estimativas.

## 1.1 MOTIVAÇÃO

Os últimos anos mostraram uma mudança abrupta no paradigma da computação pessoal. A popularidade de dispositivos móveis equipados com tecnologias de localização propiciou expansão de diversos serviços de informação ao adicionar a localização como dimensão extra para os seus recursos. Uma variedade de aplicativos e serviços utilizam coordenadas de localização no cotidiano dos seus usuários. São recursos cujas aplicações vão desde o auxílio na busca por um endereço até o compartilhamento de artefatos, como fotos (VAUGHAN-NICHOLS, 2009).

Apesar de sua robustez e alcance de transmissão, PLBs (*Personal Locator Beacons*) e outros mecanismos digitais de sinalização de emergência, são recursos de localização ainda pouco adotados frente a celulares e outros dispositivos digitais capazes de emitir sinais de radiofrequência.

Aproveitar recursos embarcados em tecnologias que já podem ser consideradas *commodities*, presentes de maneira ubíqua, com baixo custo de acesso e miniaturização de componentes, facilita a aplicação de soluções computacionais em diferentes *designs* destinados à localização.

Dispositivos emissores de sinais de radiofrequência podem ser localizados automaticamente por sistemas de busca autônomos programados adequadamente e que possuam o *hardware* necessário, como sensores compatíveis entre emissores e receptores. Isso possibilitaria o emprego de VANTs automatizados em tarefas de busca e resgate. Tomando partido do uso de aparelhos celulares e outros dispositivos pessoais, o processo de sensoriamento pode ser facilitado com a utilização de mais de um padrão de transmissão sem

fio embarcado, oferecendo maior redundância no sinal analisado.

A localização exata de um dispositivo é considerada crítica quando se trata de aplicações específicas, como reconhecimento de território inimigo, detecção de alvos e operações de socorro.

### 1.1.1 ALINHAMENTO COM A ESTRATÉGIA NACIONAL DE DEFESA

A Estratégia Nacional de Defesa (END) (DA DEFESA, 2008) é o decreto que estabelece diretrizes para a adequada preparação e capacitação das Forças Armadas, de modo a garantir a segurança do país em situações de crise.

Dentre as ações da END, a diretriz de Ciência, Tecnologia e Inovação (CT&I) tem como objetivo fomentar a pesquisa e o desenvolvimento de produtos e sistemas militares e civis que compatibilizem as prioridades científico-tecnológicas com as necessidades de defesa. O plano nacional de pesquisa e desenvolvimento de produtos de defesa, tem como escopo prioritário a busca do domínio de tecnologias consideradas estratégicas e de medidas para o financiamento de pesquisas. Além do mais, especifica a necessidade de viabilizar o desenvolvimento integrado e a conclusão de projetos relacionados à defesa nacional, com ênfase para o desenvolvimento e a fabricação, entre eles, de sistemas de aeronaves remotamente pilotadas (SARPs).

Considera-se decisiva a necessidade de preferir a opção que minimize a dependência tecnológica ou política em relação a qualquer fornecedor que possa pretender, por conta dessa participação, inibir ou influir sobre iniciativas de defesa desencadeadas pelo Brasil.

Em respeito à utilização do espaço exterior como meio de suporte às atividades de defesa, os satélites para comunicações, controle de tráfego aéreo, meteorologia e sensoriamento remoto desempenharão papel fundamental na viabilização de diversas funções em sistemas de comando e controle. As capacidades de alerta, vigilância, monitoramento e reconhecimento também podem ser aperfeiçoadas por meio de sensores, a bordo de satélites ou aeronaves remotamente pilotadas (ARPs).

O sistema integrado de Comando e Controle de Defesa deverá ser capaz de disponibilizar, em função de seus sensores de monitoramento e controle do espaço terrestre, marítimo e aéreo brasileiro, dados de interesse do Sistema Nacional de Segurança Pública. De forma recíproca, o Sistema Nacional de Segurança Pública deverá disponibilizar ao sistema de defesa nacional dados de interesse do controle de fronteiras, no que diz respeito às atividades ligadas aos crimes transnacionais fronteiriços. Fomenta-se o avanço nos programas de ARPs, para vigilância e combate. SARPs poderão se tornar meios cen-

trais no combate aéreo, além de possibilitar maior precisão no monitoramento e controle do território nacional.

Ainda no âmbito da Estratégia Nacional de Defesa, um dos objetivos estratégicos das Forças Armadas do Brasil é "A capacidade para levar o combate a pontos específicos do território nacional". Implicando a necessidade de sistemas de precisão, capazes de permitir a discriminação de alvos em situações nas quais forças nacionais poderão estar entremeadas ao inimigo. Dessa forma, podem ser destacados sistemas de localização capazes de operar frente a interferências inerentes aos sistemas GNSS.

Por fim, entre as diretrizes da END, é pautada a necessidade de ampliar a capacidade de atender a compromissos internacionais de busca e salvamento. Estabelecendo como tarefa prioritária o aprimoramento dos meios existentes e da capacitação do pessoal envolvido com as atividades de busca e salvamento em território nacional, em águas jurisdicionais brasileiras e em áreas pelas quais o Brasil é responsável, ao atender compromissos internacionais. Tal compromisso pode lançar mão da aplicação de tecnologias de localização precisas e metodologias versáteis para cada cenário.

## 1.2 CARACTERIZAÇÃO DO PROBLEMA

O resgate de vítimas em áreas de desastre é uma atividade de risco às equipes empenhadas no trabalho e que pouco podem contar com ferramentas automáticas auxiliares. Fora as dificuldades de resgate, deve-se considerar que o processo de localização de uma pessoa é de grande dificuldade e imprecisa em vários cenários, onde o próprio acesso ao local já é um fator restritivo. Entre 2013 e 2014 foram registradas 61 pessoas desaparecidas em áreas de reserva Florestal no estado de São Paulo, um aumento de 17,3% (G1, 2016c). Desastres, como o da Cidade de Mariana (Minas Gerais) em 2015, registram vários mortos e desaparecidos (G1, 2016a), onde as equipes de resgate enfrentaram dificuldade para executar a localização das vítimas e acessar o local.

Veículos não tripulados já estão em uso por diversos batalhões de bombeiros no Brasil, mas estes aparelhos ainda são operados como dispositivos de observação remota sem rotinas automatizadas de busca e localização (G1, 2016b).

Apesar da facilidade de acesso a dispositivos como *smartphones*, a estrutura de telecomunicações celular GSM (*Global System for Mobile Communications*) atualmente não pode ser considerada onipresente, existindo maior cobertura de sinal em áreas urbanas ou de trânsito de veículos; de baixa qualidade em regiões rurais e praticamente inexistente em áreas isoladas, como reservas florestais. A figura 1.1 exibe os dados de intensidade de

sinal GSM no estado do Rio de Janeiro coletados pela plataforma OpenSignal, os pontos em roxo são as amostras com alta recepção e em marrom em baixa. Um *smartphone* tem a funcionalidade consideravelmente limitada em regiões sem a infraestrutura de comunicações estabelecida, sendo impossibilitado de fazer uma ligação em caso de emergência, por exemplo. Porém, esses dispositivos contam com rádio transmissores, como WiFi e bluetooth, capazes de emitir um sinal radiofrequência que pode ser identificado dentro do seu raio de alcance.

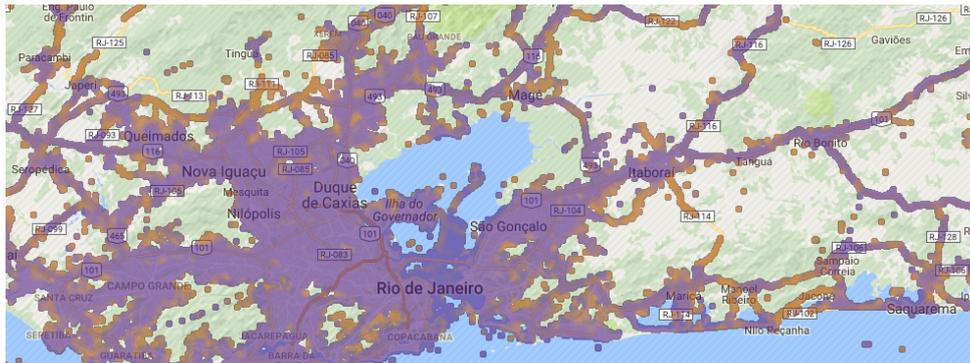


FIG. 1.1: Mapa do estado do Rio de Janeiro com a distribuição de antenas celulares GSM, os pontos em roxo se referem a amostras com alta recepção e em marrom, em baixa (Fonte: OpenSignal)

Para um VANT em rota de navegação autônoma, a detecção de ondas de radiofrequência em protocolo conhecido, e eventual determinação do local do seu emissor, consome menos recursos para localização que a utilização de outros sensores como câmeras, que dependem de técnicas de visão computacional e um considerável custo de processamento, que em *hardwares* embarcados pode ser proibitivo (YU et al., 2016).

A estimativa precisa da localização de um emissor de radiofrequência necessita de várias amostras de sinais capturados quando o ambiente não é conhecido *a priori* (ZHANG et al., 2011). Determinar um ponto com poucas verificações de sinais, ou com amostras de baixa qualidade, elevam a margem de erro no cálculo, acentuando o desvio padrão. A utilização de múltiplos sensores em paralelo pode auxiliar na determinação mais precisa e oferecer uma taxa de redundância útil à estimativa (LIN et al., 2010).

Em uma política mais otimizada de consumo de energia, são necessárias taxas de amostragem para verificação de intensidade de sinal em um limite ideal que não prejudique a precisão da determinação da localização de um alvo. Sensores diferentes possuem especificações peculiares quanto à precisão, consumo de energia e alcance de operação.

Um VANT, durante uma operação de busca e localização, muitas vezes não terá acesso a uma estação de comando e controle em solo (GCS, *Ground Control Station*).

Como exemplificado na figura 1.2, a área em rosa representa ausência de transmissão de telemetria e as áreas em azul e amarelo representam a cobertura de sinal bluetooth e WiFi, respectivamente, de um alvo que é localizado pelo veículo em sua rota de navegação autônoma. Dependendo exclusivamente de seus sistemas de computação embarcados para a estimativa do ponto de transmissão, o algoritmo utilizado deve ser consideravelmente simples de modo que possibilite a sua execução em *hardwares* com restrições computacionais (como processamento, quantidade de memória e fornecimento de energia) e não depender de bases de dados e outras fontes de informação externas.

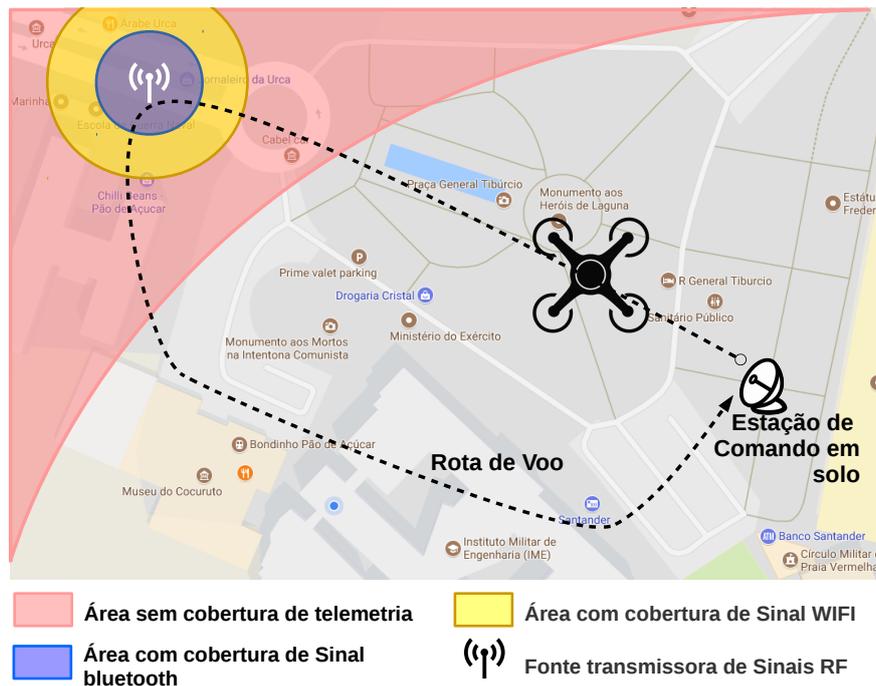


FIG. 1.2: Uma rota de voo autônoma de VANT com regiões de sinais disponíveis

Poucos trabalhos referenciam aplicações práticas para localização de ativos em ambientes externos e fora de zonas urbanas, utilizando estimativas baseadas em intensidade de sinal radiofrequência. Weber et al. (2010) explica como mapeamentos baseados em *war-driving* são mais eficientes quando aplicados em cidades, devido à quantidade de emissores de WiFi.

Alguns desafios associados à análise de registros obtidos por coletas de sinais envolvem flutuações das amostras recebidas entre diferentes *scans* e sensores. Fenômenos associados à natureza da propagação de ondas de radiofrequência como dispersão, atenuação, reflexão, refração, difração, distorção e ruído são exemplos desses desafios. Esse comportamento deve ser considerado de maneira que eventuais erros não se propaguem ao tratar problemas de estimativa.

### 1.3 OBJETIVO

Esse trabalho tem por objetivo apresentar um método de baixo custo computacional para localização automatizada de alvos transmissores de sinal de radiofrequência compatíveis com os padrões IEEE 802.11 e bluetooth de maneira precisa e dinâmica, empregando fusão de sensores de modo a oferecer maior redundância e taxa de acerto frente a técnicas que utilizam um único sensor.

A plataforma adotada deve ser acessível: componentes de *hardware* devem ser intercambiáveis e os requisitos de *softwares* precisam possuir licenças *opensource*, diminuindo assim a dependência tecnológica com soluções exclusivas.

A metodologia apresentada deve ser validada em testes de voo reais, de forma a responder às seguintes questões durante os seus ensaios:

**Questão de Pesquisa 1.** *Quais são os parâmetros a serem considerados na estimativa de posição de um alvo?*

**Questão de Pesquisa 2.** *Quais os requisitos que devem ser considerados ao utilizar rádios no padrão bluetooth como sensores de localização?*

**Questão de Pesquisa 3.** *Quantos pontos de amostragem de sinal podem ser considerados como o mínimo para uma estimativa razoável de localização de um ativo?*

### 1.4 CONTRIBUIÇÕES ESPERADAS

Partindo da premissa que a técnica de *wardriving* pode ser empregada como recurso para a localização automatizada e de que a análise de sinais de radiofrequência emitidos por antenas bluetooth podem gerar maior precisão e velocidade na estimativa, as contribuições esperadas para o presente trabalho são:

- a) Apresentar uma metodologia extensível para localização por radiofrequência;
- b) Propor um *framework* escalar, capaz de comportar múltiplos sensores;
- c) Desenvolver um algoritmo para localização, implementável em computadores de bordo de baixo custo para aeronaves não-tripuladas;
- d) Adotando técnicas de fusão de sensores, colaborar com outras linhas de pesquisa em algoritmos de localização; e
- e) Determinar os recursos tecnológicos necessários para localização autônoma de ativos.

## 1.5 ORGANIZAÇÃO DA DISSERTAÇÃO

O conteúdo dessa dissertação está estruturado na seguinte sequência: a próxima seção apresenta a fundamentação teórica do trabalho e os desafios sobre o problema de estimativa de localização. Na seção 3, é apresentada a metodologia proposta, sua arquitetura e definições sobre o VANT e outras tecnologias adotadas. O 4º capítulo dessa dissertação apresenta o planejamento dos diferentes tipos de ensaios executados e a análise dos resultados experimentais obtidos. Na seção 5, são feitas as conclusões e propostas sobre trabalhos futuros que podem partir da metodologia. Junto à dissertação, são entregues 6 anexos e 2 apêndices que explicam detalhadamente a especificação e os processos de montagem, configuração e procedimentos de uso da aeronave e dos sistemas embarcados utilizados no processo experimental.

## 2 TÓPICOS TUTORIAIS

Os conceitos de radiofrequência, e os fenômenos inerentes à tecnologia, soluções de localização baseadas em rádios WiFi (IEEE 802.11), bluetooth, GSM e GPS bem como as definições nacionais de VANT, técnicas de *wardriving* e dos algoritmos de fusão de sensores serão apresentados neste capítulo. Ao final, trabalhos relacionados serão referenciados e comentados.

### 2.1 RADIOFREQUÊNCIA

São ondas eletromagnéticas que abrangem as faixas entre 3 KHz (quilohertz) e 300 GHz (gigahertz) e que em geral carregam informação. Sua categorização depende da frequência e do comportamento da onda. Legalmente sinais de radiofrequência devem ocupar intervalos bem definidos de frequência (DE TELECOMUNICAÇÕES, 2016). Nesse intervalo, diversas tecnologias de redes de comunicação sem fio de uso cotidiano se encontram, como as utilizadas na especificação *wireless* IEEE 802.11 (ondas UHF de 2,4GHz, com 12cm de amplitude, até 5GHz, com 6cm de amplitude) e bluetooth (ondas UHF de 2,4 até 2,485GHz) (S.; MILLER, 2008).

O comportamento de um sinal de radiofrequência está diretamente associado ao meio de propagação, podendo haver interferência de diversas naturezas, como atmosférica, reação a superfícies e até mesmo outras ondas eletromagnéticas (SIQUEIRA, 2017). Podem-se citar como fenômenos relevantes à propagação do sinal:

- a) **DISPERSÃO:** É o fenômeno que acontece quando uma onda, que é resultante de várias outras, entra em um novo meio onde a velocidade de propagação é diferente para cada uma de suas componentes, fazendo com que o espectro se divida nesse novo meio;
- b) **ATENUAÇÃO:** Quando há uma diminuição da amplitude e distância de alcance da onda após entrar em um novo meio, onde mais energia é necessária para continuar a propagação;
- c) **REFRAÇÃO:** Ocorre quando uma onda sofre mudanças na direção de sua propagação, quando passa de um meio para o outro;

- d) DIFRAÇÃO: Trata-se do encurvamento sofrido por uma onda quando ela encontra obstáculos a sua propagação;
- e) REFLEXÃO: Definido como o fenômeno que ocorre quando uma onda é refletida ao incidir em uma superfície.
- f) DISTORÇÃO: Uma influência capaz de modificar os componentes (frequência, fase ou amplitude) originais de um sinal.
- g) RUÍDO: Qualquer tipo de erro que não está presente no sinal original (transmissor), mas perceptível ao receptor, podendo este ser de origem interna ou externa ao meio de transmissão. Um ruído pode ser de natureza aleatória ou determinística.

Fenômenos dessa natureza levam a duas características de grande interesse para a área de localização por radiofrequência: Desvanecimento de percurso (*path loss*), quando há uma redução de densidade da potência de uma onda eletromagnética pelo meio de transmissão; e o efeito de multi-percurso (*multipath*), comportamento que resulta na chegada do mesmo sinal ao receptor em tempos diferentes.

## 2.2 WIRELESS SENSOR NETWORK

*Wireless Sensor Networks* (WSNs) são redes constituídas por sensores autônomos espacialmente distribuídos. Esses dispositivos são capazes de monitorar condições ambientais ao seu redor, tais como temperatura, pressão, ondas eletromagnéticas e som de um local. Para este trabalho, o principal emprego de *wireless sensor network* encontra-se na identificação e localização de ativos emissores de rádio frequência num meio.

### 2.2.1 GPS (E OUTROS SISTEMAS DE LOCALIZAÇÃO POR SATÉLITE)

O sistema de radiofrequência comumente utilizado para a determinação do posicionamento é o GPS. Inicialmente financiado e desenvolvido pelo governo dos Estados Unidos da América na década de 70, é composto por uma rede de 24 satélites orbitando ao redor da Terra. A tecnologia original permite determinar a localização em qualquer lugar do planeta com precisão de aproximadamente 10 metros. Para determinar a sua posição, o receptor GPS precisa obter sinais de ao menos 4 satélites diferentes. Quatro satélites representam 4 esferas definidas pela diferença entre o tempo de transmissão (do satélite) e o tempo de recepção (do receptor GPS), a sua interseção define a localização.

Sistemas apoiados por outras nações, como o Glonass, BeiDou e Galileo, também podem ser considerados como soluções de posicionamento via satélite, uma vez que os seus princípios de funcionamento são equivalentes ao do sistema GPS (SAPIEZYNSKI et al., 2015). Num contexto geral, independentemente do sistema satelital empregado, utiliza-se o termo GNSS (Global Navigation Satellite System) para definir a técnica.

A maior parte dos trabalhos originais sobre aplicações sensíveis ao contexto de localização eram focadas na utilização de GPS e informações de células de transmissão GSM. Apesar do GPS apresentar boa precisão, é uma solução de baixa eficiência energética. (BYCHKOVSKY et al., 2006) (NICHOLSON; NOBLE, 2008)

Outro problema conhecido é que sinais dos satélites de GPS não podem ser utilizados em ambientes *indoor* (dentro de edificações), levando à adoção de outros sinais de transmissão, como o IEEE 802.11, que são mais presentes em áreas urbanas. Localização por trilateração de torres de celular (GSM, por exemplo) podem ser utilizadas por aplicações, mas sua imprecisão reduz a sua usabilidade em diversos contextos.

### 2.2.2 REDES GSM

Telefones celulares e *smartphones* ganharam muita popularidade nos últimos anos e são comumente utilizados na vida cotidiana. A maioria dos aparelhos utilizados atualmente se comunicam por uma rede digital GSM (*Global System for Mobile Communications*). Redes GSM são redes celulares em que cada célula tem a sua própria estação rádio base (ERB), cada uma com um identificador único. Uma vez que o telefone conhece a célula na qual está conectado, pode-se utilizar essa informação para determinar a sua posição. Porém, essa solução tem um grande defeito: a sua precisão. Enquanto em áreas urbanas as distâncias entre estações são de 200 a 500 metros, essa diferença escala para alguns quilômetros em áreas rurais degradando a estimativa da posição. A precisão pode ser aumentada pela trilateração de sinal calculado pelo aparelho entre diferentes torres celulares em suas cercanias, mas esta também tende a diminuir em regiões com menor densidade populacional (ANANTHANARAYANAN; STOICA, 2009).

Outra observação importante é que em diversas regiões isoladas, como reservas florestais, desertos e áreas inabitadas, a presença de infraestrutura de comunicação é muitas vezes inexistente fazendo com que aplicações que necessitem de localização utilizem outros meios (ver figura 1.1).

### 2.2.3 CARACTERÍSTICAS DO PADRÃO IEEE 802.11

As redes sem fio WiFi (IEEE 802.11) ganharam muita popularidade nos últimos anos e podem prover um meio para determinar a localização, especialmente em áreas residenciais e nas dependências de grandes organizações. Para o acesso aos dados de uma rede, clientes WiFi se conectam a pontos de acesso (APs, *Access Points*, do original em inglês). Esses aparelhos podem ser identificados por um nome atribuído (SSID) ou endereço de *hardware* (*Mac Address*) associado ao AP. Para determinar as melhores taxas de transmissão, os clientes medem a intensidade (RSSI) e a qualidade do sinal recebido. Esses parâmetros também possibilitam a determinação de distância até o ponto de acesso. Diversas aplicações sensíveis ao contexto estimam a posição de um usuário, de posse de um dispositivo móvel, ao verificar a intensidade de sinal de diferentes *Access Points* nas cercanias. (ANANTHANARAYANAN; STOICA, 2009).

Atualmente, entre as conexões disponíveis em dispositivos móveis, transmissores WiFi provém a melhor qualidade de banda de dados, alcance e custo benefício energético para conexões de dados. Por outro lado, a sua eficiência em estado de *idle* (sem a comunicação ou conexão estabelecida) provoca um grande desperdício energético, em especial durante a fase de escaneamento por novas redes. Técnicas de otimização do protocolo até trabalhos que utilizam vários rádios transmissores sem fio em um único dispositivo (LIN et al., 2010) foram estudados como forma de melhorar o consumo de energia desse padrão.

Ao longo de seus ciclos de desenvolvimento, diversas versões do padrão IEEE 802.11 foram criadas apresentando otimizações em suas especificações, como visto na tabela 2.1. Mudanças como banda de dados e, em especial, nos atributos de frequência e potência máxima de transmissão, são características que afetam diretamente o seu raio de alcance e, por consequência, sua aplicação em sistemas de localização em WSN.

TAB. 2.1: Diferenças entre as especificações comerciais do padrão IEEE 802.11

Padrão IEEE	Freq. (Ghz)	Banda (Mbps)	Potência (mW)	Alcance (m)
802.11a	5	54	100	35
802.11b	2,4	11	100	35
802.11g	2,4	54	100	70
802.11n	2,4 e 5	600	100	70
802.11ac	5	1300	160	35

## 2.2.4 CARACTERÍSTICAS DO PADRÃO BLUETOOTH

O padrão Bluetooth foi desenvolvido em 1999 como uma tecnologia substituta aos meios de transmissão por cabo. O alcance nominal de dispositivos Bluetooth é de apenas alguns metros (tabela 2.2), mas versões mais recentes da especificação consideram alcances com distâncias equivalentes ao padrão WiFi (SIG, 2016).

TAB. 2.2: Especificações de largura de banda e distância de transmissão de cada versão do padrão bluetooth (Fonte: SIG (2016))

Versão	Velocidade Máx.	Alcance Máx. (m)
1.1 e 1.2	723,1 Kbps	10 a 20 (100m classe 1)
2.0	2,1 Mbps	5 a 10
2.1	3 Mbps	5 a 10
3.0	24Mbit/s	10
4.0	25Mbit/s	60,96
5.0	50Mbit/s	444 (classe 3)

Apesar destas limitações, é uma tecnologia que tem aplicabilidade em sistemas de localização podendo prover estimativas de posicionamento com boa granularidade. Em (Y. YONEYAMA; SHINODA, 2002), os autores investigam a usabilidade do padrão bluetooth em sistemas de localização.

Como o padrão bluetooth foi desenvolvido para a criação de PAN (*Personal Area Networks*) e é tipicamente de baixo alcance, sua utilização em sistemas de localização é restrita a distâncias menores em comparação às técnicas que utilizam o padrão WiFi (KREVL; CIGLARIC, 2006).

O padrão bluetooth apresenta também maior eficiência energética frente ao padrão WiFi e mesmo assim é ignorado por diversas soluções de localização. Durante os últimos anos, o padrão bluetooth está em mais dispositivos embarcados do que transmissores no padrão IEEE 802.11 (HERFURT; MULLINER, 2004).

Dispositivos móveis atuais como *smartphones* são equipados com interfaces múltiplas de rede, incluindo rádios WiFi com mais de um canal, interfaces bluetooth e GSM (tabela 2.3). Essas interfaces seguem padrões diferentes, porém podem ser consideradas complementares em fatores como *throughput*, alcance e consumo de energia (RAHMATI; ZHONG, 2007) (PERING et al., 2006).

O padrão *Bluetooth Low Energy* (BLE) foi criado para oferecer melhor autonomia energética aos dispositivos. Essa tecnologia possibilitou a utilização de fontes de alimentação de até 3 Volts, expandindo consideravelmente o tempo de operação em relação aos padrões bluetooth anteriores (FARAGHER; HARLE, 2015). O padrão BLE oferece

TAB. 2.3: Comparação de largura de banda e consumo médio de energia de diferentes estados entre dispositivos GSM, WiFi e bluetooth

	Banda de dados	Transferência (J/MB)	Idle(W)	Procura(W)	Alcance(m)
GSM (3G)	Menos de 1Mbps	100	-	-	500
WiFi	54Mbps	5	0,77	1,29	100
Bluetooth 4.0	25Mbps	0,1	0,01	0,12	10

quatro modos distintos: periférico, central, difusão e observador. Os modos periférico e difusão, constituem os principais modos de operação para localização de ativos. Os dispositivos BLE buscam periodicamente por transmissões, o sinal capturado consiste em pacotes de dados que identificam cada transmissor e são padronizados por protocolos de encapsulamento, como o iBeacon da Apple e Eddystone da Google (G. BRANCO; BORDIM, 2017).

### 2.3 DISPOSITIVOS DE LOCALIZAÇÃO DE EMERGÊNCIA

Também conhecidos como transmissores de localização de emergência, são dispositivos que enviam um sinal de alerta (*distress signal*) para uma infraestrutura especializada que integra um sistema completo de busca e resgate. Segundo a *National Oceanic and Atmospheric Administration* dos EUA (NoAA) (OCEANIC; ADMINISTRATION, 2016), estes sistemas são sub categorizados como:

- EPIRB (*Emergency position-indicating radio beacons*) - para uso náutico;
- ELTs (*Emergency Locator Transmitters*) - para uso na aviação;
- PLBs (*Personal locator beacons*) - para aplicações em solo.

Outros Padrões de transmissores de sinais de emergência incluem:

- SART (*Search and rescue transponder*) - Busca de aeronaves por radares;
- ENOS (*Elektronisches Notruf und Ortungssystem*) - usado por mergulhadores.

Os EPIRBs são transmissores de localização usados em situações de emergência, operados através do consórcio de satélites (COSPAS-SARSAT). Quando o dispositivo é ativado, este aparelho envia sinais intermitentes com dados que possibilitam a localização de pessoas, embarcações ou aeronaves necessitando de resgate.

Suas principais características são:

- Dispositivos voltados à sinalização de emergência para uso náutico;
- Resistentes e a prova d'água;
- Possibilitam a ativação manual ou automática;
- Utiliza a frequência de 406 MHz (*distress frequency*);
- Comunicação direta por satélite (operados pelo Consórcio Cospas-Sarsat);
- Envio de informações da embarcação e a sua localização, dependendo do modelo;
- Autonomia de bateria de 24 a 48 horas;
- Dependente de um sistema de resposta a incidentes para o resgate a nível global.

De um modo geral, atualmente a maioria dos transmissores de localização de emergência trabalham na faixa de 406 MHz e são operados pela mesma rede de satélites do Consórcio Cospas-Sarsat.

Apesar de sistemas especializados de localização em caso de emergência estarem disponíveis há mais de uma década, em diversas atividades esses dispositivos especializados ainda são negligenciados, ao contrário de computadores pessoais e *smartphones*. Segundo Munshi et al. (2014), tais dispositivos precisam de soluções práticas e mais acessíveis para atender a um público maior e mais diversificado.

## 2.4 LOCALIZAÇÃO *INDOOR* E *OUTDOOR*

Sistemas de localização são altamente sensíveis ao meio e à escolha correta dos sensores que serão utilizados para a determinação de posicionamento, dependendo diretamente do ambiente onde a aplicação será executada.

Trabalhos sobre sistemas de localização dividem os casos de uso em dois meios: *Indoor* e *Outdoor*.

Meios tipo *outdoor*, ou externos, são menos susceptíveis à obstrução de sinais de radiofrequência e, em geral, podem contar com um bom sinal dos receptores de GPS para a estimativa de sua localização. Aplicações de *wardriving*, por exemplo, podem lançar mão dessa precisão para inferir o posicionamento de *beacons* de radiofrequência apenas analisando a força do sinal recebido, comparando com a própria localização exibida pela interface GPS. Trabalhos apresentados em Kim et al. (2006) sugerem propostas de múltiplos sensores para melhorar a estimativa de localização outdoor em meios urbanos.

Dentro de edificações, ou meios tipo *indoor*, não é possível contar com sinais de GPS para os sistemas de posicionamento. Desta maneira, outros sensores devem ser levados em consideração.

Sinais de redes sem fio são comuns hoje em dia e são utilizados em diversas aplicações de estimativa de posicionamento *indoor*, como shopping centers ou aeroportos (NEISHABOORI; HARRAS, 2013). Para aumentar a precisão, algumas técnicas empregam sensores inerciais em *smartphones* para inferir uma posição dentro de um local a partir de um ponto conhecido, enquanto outras utilizam sensores ambientais extras aos *beacons* WiFi melhorando a estimativa (PRIYANTHA, 2005).

Sinais de radiofrequência em ambientes *indoor* são suscetíveis a interferências do meio e novos modelos devem ser estudados para analisar a propagação das ondas e melhorar a precisão (YU et al., 2016).

Para prover uma cobertura generalizada, redes sem fio no padrão IEEE 802.11 são comumente utilizadas em técnicas de localização. O problema na dependência de posicionamento por WiFi é que os pontos de acesso devem ser conhecidos a priori. O mapeamento desses *access points*, ou APs, por técnicas de *wardriving*, é feito constantemente por grandes empresas e suas bases são atualizadas regularmente; mas, devido a questões relativas a propagação do sinal, a estimativa de sua localização pode ser diferente da realidade (SAPIEZYNSKI et al., 2015).

Ao longo de uma década, diversas técnicas de localização por radiofrequência foram sugeridas, desde as que envolvem assinatura de RSSI (intensidade do sinal recebido) como a RADAR e HORUS, com resolução de 3 a 5 metros, até métodos que oferecem maior precisão, de até 1 metro, como a CompAcc, EZ e UNLOC (NEISHABOORI; HARRAS, 2013).

A estimativa de posicionamento baseado em WiFi, GSM ou bluetooth possui uma importante vantagem frente aos sistemas de localização por satélite: além de atenderem o caso de uso de ambientes externos, estes métodos também possibilitam a orientação em ambientes *indoor*, dentro de edificações. Por outro lado, sistemas como o GPS possuem uma ótima performance em ambientes abertos, como em áreas rurais onde há escassez de transmissores de radiofrequência.

## 2.5 LOCALIZAÇÃO POR *WARDRIVING*

Originalmente definido como a técnica de procura por redes WiFi através de veículos automotivos equipados com sensores (THOMAS KING, 2006), atualmente o termo *Wardriving*

é utilizado para qualquer busca e localização por pontos de acesso a redes sem fio. São considerados outros meios de transporte, como a pé, de bicicleta e até utilizando animais de estimação (BRANSFIELD, 2014). De fato, como mencionado em (TSUI et al., 2010) a qualidade do mapeamento de pontos de acesso pode diferenciar drasticamente durante o processo de *wardriving*, dependendo da velocidade do sistema de localização.

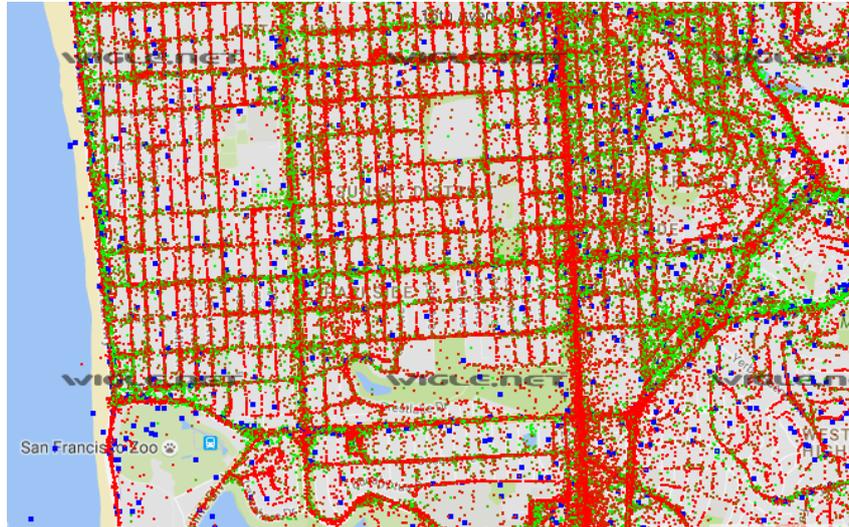


FIG. 2.1: Mapa de São Francisco (EUA) com todos os *Access Points* (APs) reconhecidos pelo Projeto WiGLE (Wireless Geographic Logging Engine) em 11/2016. Os pontos em vermelho demonstram a localização de APs com alta precisão (devido ao número de amostras obtidas) e em verde a posição de APs com baixa precisão. Em azul são considerados APs ainda sem confirmação de localização, ou errantes. (licença CC BY-SA)

Sistemas sofisticados de comunicação, como os presentes em *smartphones*, se tornaram ubíquos nos últimos anos e um interesse crescente no chamado Sensoriamento Participativo, ou *Crowdsourcing*, surgiu. Por este processo é possível obter, e manter atualizada, uma grande massa de dados sobre o mapeamento de *access points* em nível global, através de técnicas de *wardriving* executadas por aparelhos celulares e outros dispositivos móveis de uso cotidiano conectados à internet (RAI et al., 2012) (KIM et al., 2012). A figura 2.1 mostra um fragmento da massa de dados coletada pelo sistema de *Crowdsourcing* Wigle, exibindo todos os pontos de acesso WiFi descobertos em uma região da cidade de São Francisco (EUA) em novembro de 2016.

### 2.5.1 PRINCIPAIS FERRAMENTAS PARA *WARDRIVING*

Algumas ferramentas *open source* específicas para *wardriving* são comumente adotadas em projetos de localização de redes sem fio em tempo real, como o Kismet (WIRELESS,

2016) e o Airdump-NG (AIRCRACK-NG, 2016), essas ferramentas também são capazes de coletar pacotes de dados e radiofrequência que a interface de rede WiFi detecta. Essa técnica também é capaz de descobrir APs ocultos ao observar clientes legítimos se comunicando com seus *beacons* (JENKS, 2014).

Ferramentas passivas como Kismet e Netsh (uma alternativa ao Netstumbler) são desenvolvidas com o intuito de trabalhar na detecção de redes sem fio de maneira não intrusiva, mas obtendo dados suficientes sobre o estado de seu rádio e configurações básicas de segurança (EKHATOR, 2010).

Para rastreamento de transmissores bluetooth, podem ser usadas aplicações como o Bluediving (SOURCEFORGE, 2007), um conjunto de ferramentas para ataques em dispositivos no padrão e o BtCrawler (MUELLER, 2016), uma aplicação para o sistema operacional Android capaz de utilizar os recursos de um *smartphone* para detectar dispositivos bluetooth próximos. A ferramenta Kismet, além de operar para redes no padrão IEEE 802.11, também pode localizar *beacons* bluetooth.

## 2.6 FUSÃO DE SENSORES

Devido à natureza de suas amostras, certas aplicações recorrem ao uso de múltiplas fontes de dados para otimizar as suas estimativas. Sistemas de localização podem utilizar mais de um sensor para melhorar a sua precisão. As técnicas de fusão de sensores (*Sensor Fusion*, em inglês) empregam algoritmos específicos para tratar dados de diferentes fontes de modo a obter melhores resultados. Essa técnica combina a estimativa de posição obtida por diversos dispositivos ou sensores. A ideia principal da solução é que os resultados gerados por um algoritmo de fusão de sensores sejam mais eficientes do que a posição estimada por apenas uma fonte.

Em muitos casos, influenciado pelo ambiente, os dados de um sensor podem conter ruídos e levar a uma estimativa errônea de localização e dispositivos diferentes podem até mesmo apresentar resultados conflitantes (CRIVELLA, 2015).

As técnicas de fusão de sensores prometem resolver os problemas de conflitos entre resultados de diferentes fontes e podem ser utilizadas para aumentar a precisão. Thomas King (2006) mostra como pode ser minimizado o número de consultas efetuadas a um sensor, ao mesmo tempo que é obtido um nível de precisão mais satisfatório, através de um algoritmo de rede Bayesiana. O projeto Place Lab da Intel utilizou pontos de acesso WiFi, torres de celular GSM e algoritmos de *sensor fusion* baseados em Filtro de Partículas para otimizar a precisão e a cobertura de seu sistema de localização *indoor*

(KREVL; CIGLARIC, 2006).

**Filtros de Kalman:** baseado na utilização de um estimador de estados que presume o valor de uma variável a partir de entradas com ruído, expressado por um algoritmo que se fundamenta no histórico das entradas. O filtro de Kalman assume um modelo linear, sendo também seus estados e estimativas transformações lineares. Dentro de um instante  $t$ , o estado  $x_t$  é definido como a combinação do estado anterior  $x_{t-1}$  dado uma matriz de transformação  $A$ , um controle  $u_t$  junto também de uma matriz de transformação  $B$ . Sendo  $E$  o ruído proveniente do sistema. Pode-se tornar o modelo mais simples ao desconsiderar a variável de controle  $u$  e utilizar  $A$  como matriz identidade (Bulten, 2015).

$$x_t = A_t x_{t-1} + B_t u_t + E_t \approx x_{t-1} + E_t \quad (2.1)$$

Sendo  $C$  a matriz de transformação e  $\delta$  o ruído, define-se como um estado  $X$  resulta em uma medição  $Z$ :

$$Z_t = C_t x_t + \delta_t \quad (2.2)$$

O momento de atualização do filtro pela predição, ocorre seguindo a equação:

$$\begin{aligned} \bar{\mu}_t &= \mu_{t-1} \\ \bar{\Sigma}_t &= \Sigma_{t-1} + R_t \end{aligned} \quad (2.3)$$

Em que  $\mu$  representa a predição e  $X$  o estado real, já  $\Sigma$  define a certeza da predição e  $R$  o ruído causado pelo sistema. Supõe-se que devido o ambiente estático o ruído esperado procede exclusivamente do ambiente (G. BRANCO; BORDIM, 2017). Em ambientes de coleta dinâmicos, onde há o movimento do sensor durante a coleta de sinais, deve-se aumentar o valor dessa variável. A partir da etapa de predição, registra-se o ganho de Kalman como:

$$K_t = \bar{\Sigma}_t (\bar{\Sigma}_t Q_t)^{-1} \quad (2.4)$$

Sendo  $Q_t$  o ruído da informação. Para modelagem sobre problemas de RSSI, a variância simboliza o valor de influência. Desta maneira, o ganho juntamente a  $Q$  serve como peso para avaliar a certeza da predição contra a certeza da medição. O que torna a etapa de atualização completa por:

$$\begin{aligned}\mu_t &= \bar{\mu}_t + K_t(z_t - \bar{\mu}_t) \\ \Sigma_t &= \bar{\Sigma}_t - (K_t \bar{\Sigma}_t)\end{aligned}\tag{2.5}$$

Sendo assim, quanto maior o ganho  $K$ , mais a medição se tornará integrada ao sistema, e quanto menor este ganho for, mais a predição se torna integrada.

## 2.7 LOCALIZAÇÃO POR RADIOFREQUÊNCIA

Diversos algoritmos como FILA (WU et al., 2012), SpotFi (KOTARU et al., 2015), o filtro Alpha (PRIETO et al., 2012) e o RMCL (ZHU et al., 2013), foram propostos como técnicas de localização com o objetivo de mapear as redes WiFi. Poucos trabalhos referenciam aplicações práticas para localização de ativos em ambientes externos e fora de zonas urbanas, utilizando estimativas baseadas em intensidade de sinal radiofrequência. Weber et al. (2010) explicam como mapeamentos baseados em *wardriving* são mais eficientes quando aplicados a cidades, devido à quantidade de emissores no padrão WiFi.

A localização precisa de ativos por intermédio de aplicações de baixo custo computacional é o tema de trabalhos em desenvolvimento. Algumas propostas como o GLANSER (HAWKINSON et al., 2012) buscam estimar a posição de bombeiros em ambientes *indoor*, dentro de edificações em chamas, onde o sinal de sistemas de localização por satélite é ineficiente. Técnicas acessíveis para mapeamento de pontos de acesso WiFi, como o *wardriving*, são aplicadas de diferentes maneiras promovendo estimativa de localização mais veloz e precisa em ambientes *indoor* e *outdoor*, como uma alternativa viável aos sistemas de localização por satélite (HAEBERLEN et al., 2004).

A detecção de ondas de radiofrequência e a determinação da origem do seu emissor podem ser implementadas de maneira simplificada. Segundo Yu et al. (2016), essa abordagem é uma alternativa a soluções que empregam câmeras, que dependem de técnicas de visão computacional que podem gerar custo de processamento proibitivo em hardwares embarcados. Essas aplicações comumente recorrem a computadores remotos para resolver o problema de localização de um robô (DUNKLEY et al., 2014). O trabalho de von Stumberg et al. (2016) apresenta um sistema de navegação autônomo de baixo custo computacional, para ser embarcado em *hardwares* dentro das restrições de peso, consumo de energia e desempenho.

A eficiência energética em sistemas de localização é abordada pelos autores do *Blue-fi*, Ananthanarayanan e Stoica (2009), uma solução desenvolvida para monitorar o ambiente por meio da interface bluetooth e da triangularização de torres GSM, habilitando as

funções de WiFi apenas quando necessário. Cheng e Hsiu (2015) desenvolvem um algoritmo de programação dinâmica para otimizar consumo de energia por intermédio de agendamentos para utilização de sensores na estimativa de localização. Em Neishaboori e Harras (2013), para contornar o problema de eficiência energética, os autores abordam a medição por aglomeração de alvos, baseando-se em padrões de deslocamento, e não no comportamento de um indivíduo.

Como exibido na figura 2.2, um sistema de localização de dispositivos em uma rede de sensores sem fio (WSN) pode abordar o problema da distância entre o emissor e receptor por diferentes técnicas: Força de Sinal Recebido (RSSI), Tempo de Chegada ou Tempo Diferencial de Chegada (ToA ou TDoA, da sigla original em inglês) ou Ângulo de Chegada (AoA). Em posse da estimativa de distância ou um valor referencial que aponte esse valor proporcionalmente, deve-se utilizar uma técnica de determinação de posicionamento: lateração (ao utilizar valores absolutos de distância em diferentes pontos de coleta) ou angulação (onde a posição do alvo é determinada pela diferença entre ângulos de observação).

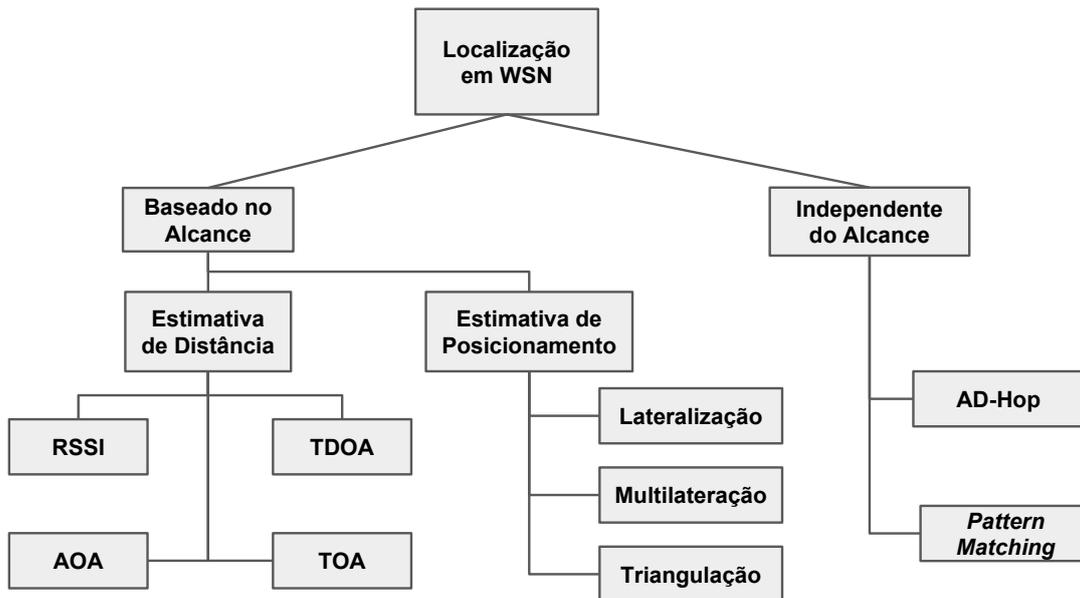


FIG. 2.2: Compartimentação do problema de localização em WSN.

As técnicas baseadas em Tempo de Chegada (ToA - *Time of Arrival*), são capazes de estimar a distância de um transmissor no mesmo padrão, utilizando o tempo de propagação do sinal até o sensor. Para tal, se faz necessário o conhecimento prévio da velocidade de propagação da onda de radiofrequência no meio utilizado. Essas técnicas também demandam que todos os nós que compõem o sistema de localização estejam com

seus relógios sincronizados.

TDoA (*Time Difference of Arrival* ou diferença de tempo de chegada) é uma técnica de localização que utiliza a diferença entre os tempos de recebimento de sinal enviado por um emissor e o tempo de sua transmissão ao receptor, ao invés de utilizar o tempo absoluto de chegada em um dos nós.

Por utilizar o tempo como atributo principal, técnicas de tempo de chegada tem uma resistência maior às distorções eletromagnéticas causadas por desvios de sinal e multipercurso (*multipath*), sendo capazes de desconsiderar casos em que o mesmo sinal refletido seja recebido mais de uma vez pelo sensor. Em contrapartida, necessitam de *hardwares* específicos ou de alteração no protocolo de comunicação nos padrões de rádio existentes (GÜNAY; ÇAVDAR, 2014).

Várias soluções em redes de sensores sem fio utilizam mecanismo de *Angle of Arrival* (AoA). Funções como rastreamento, que necessitam de monitoramento aprimorado e orientação do alvo em um ângulo de visão. Sensores que podem ser suportados incluem antenas direcionais em bases articuláveis ou uma matriz de antenas, diminuindo as interferências ambientais (de outros dispositivos de radiofrequência, por exemplo). Com o padrão de angulação da transmissão capturada, um único rádio receptor, com um sistema de múltiplas antenas, pode ser utilizado de maneira eficiente para calcular o ângulo de chegada de um sinal e a consequente determinação de sua origem (CHEN et al., 2006).

Triangulação é o método geométrico que possibilita o cálculo de posição de um elemento no espaço, baseado nos ângulos formados entre esse e outros elementos referenciais onde a posição é conhecida à priori. Em WSN, as técnicas de AoA utilizam triangulação como solução para a estimativa de posicionamento de um alvo.

A concepção básica de sistemas de estimativa de distância por RSSI está na relação entre a potência transmitida e recebida de sinais entre o nó sensor e o transmissor. A equação 2.6, indica essa relação:

$$P_r = P_t * \left(\frac{1}{d}\right)^\alpha \quad (2.6)$$

onde  $P_r$  é a potência do sinal *wireless* recebido,  $P_t$ : potência de transmissão do sinal,  $d$ : distância entre emissor e receptor, e  $\alpha$ : variável ambiental de propagação. Ao adicionar  $10 \log$  à cada lado, obtemos a equação 2.7, onde  $10 \log P_r$  é a expressão de potência convertida para dBm e os valores de  $A$  (ou  $10 \log P_t$ ) e  $\alpha$  descrevem a força de sinal recebida (RSS) e a distância de transmissão (ABDULQADER HUSSEIN et al., 2015).

$$10 \log P_r = A - 10\alpha \log d \quad (2.7)$$

O Modelo *log-normal shadow* é o mais apropriado para aplicações em redes de sensores sem fio devido a sua natureza universal e por ser configurável para diferentes ambientes. A equação 2.8 é a de *Log-distance path loss model*, onde  $\mathbf{PL}(\mathbf{d})$  se refere à potência de sinal recebida do sensor *wireless* no local  $\mathbf{d}$  da realização da amostra.  $\mathbf{P}(\mathbf{d}_0)$  é a queda de sinal perceptível em espaço livre (usualmente considerada a 1 metro de distância). Nesse modelo,  $\alpha$  se refere ao expoente de *pathloss* (perda de sinais) enquanto o parâmetro  $\mathbf{X}_\theta$  está relacionado ao efeito de sombreamento, refletindo em atenuação (em decibel).

$$PL(d) = P(d_0) - 10\alpha \log \frac{d}{d_0} + X_\theta \quad (2.8)$$

A solução para localizar um objeto no espaço depende de diversas medições tiradas em posições diferentes. Considere um alvo na posição  $P$  e medições realizadas por sensores na posição  $L$ , dentro do mesmo sistema de referência. O sensor em  $L$  não pode inferir diretamente a localização de  $P$ , mas sim estimá-la a partir da distância  $d$  calculada.

Com apenas um ponto de referência  $L$  não é possível identificar exatamente a posição de  $P$ . Cada ponto que está a uma distância  $d$  de  $L$  é um candidato em potencial para a localização do alvo. Dessa maneira, com apenas um ponto de coleta de amostras, a posição  $P$  está limitada ao círculo de raio  $d$  ao redor de  $L$ , como exibido na Fig. 2.3.

Uma estimativa mais precisa poderia considerar dois pontos de coleta,  $L_1$  e  $L_2$  com as distâncias  $d_1$  e  $d_2$  tomadas a partir de suas posições, respectivamente. Com essas duas medições de mesma importância, restringe-se as possibilidades de estimativa de posicionamento do alvo (em cinza), para até dois pontos, representado pelas interseções dos círculos de raio  $d_1$  e  $d_2$  na figura 2.3.

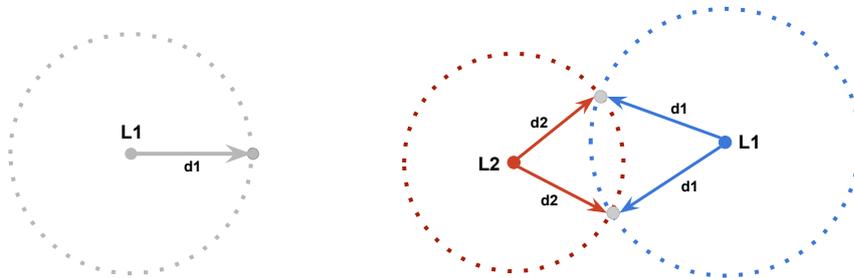


FIG. 2.3: Pontos de coleta insuficiente para trilateração

Para uma precisão absoluta, será necessário um terceiro ponto distinto de coleta,

$L_3$ . Dessa maneira o ponto de interseção entre os 3 círculos formados pelo raio de suas distâncias calculadas denotará a posição estimada P do alvo (vide Fig. 2.4).

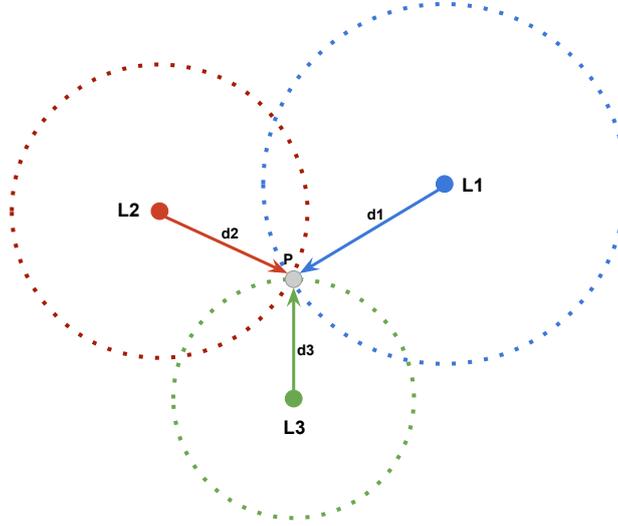


FIG. 2.4: Ponto exato de estimativa de posição por trilateração

Pode-se convencionar a interpretação matemática da posição da seguinte forma: Um ponto  $(x, y)$  no plano cartesiano está em um círculo de raio  $r$  com centro em  $(c_x, c_y)$  se e somente se for uma solução para a equação:

$$(x - c_x)^2 + (y - c_y)^2 = d_1^2 \quad (2.9)$$

Seguindo o mesmo princípio, podem-se derivar as equações para os círculos gerados por cada ponto de coleta de amostras. Cada um em sua posição expressada por coordenadas de latitude e longitude, por exemplo:  $(\lambda_1 - \phi_1)$ ,  $(\lambda_2 - \phi_2)$  e  $(\lambda_3 - \phi_3)$ , respectivamente.

Dessa maneira, o problema de trilateração pode ser resolvido matematicamente achando o ponto  $P = (\lambda - \phi)$  que simultaneamente satisfaz as equações dos 3 círculos:

$$\begin{aligned} (\lambda - \lambda_1)^2 - (\phi - \phi_1)^2 &= d_1^2 \\ (\lambda - \lambda_2)^2 - (\phi - \phi_2)^2 &= d_2^2 \\ (\lambda - \lambda_3)^2 - (\phi - \phi_3)^2 &= d_3^2 \end{aligned} \quad (2.10)$$

## 2.8 VEÍCULOS AÉREOS NÃO TRIPULADOS

Um Veículo Aéreo Não Tripulado (VANT), ou ARP (Aeronave Remotamente Pilotada), ou ainda RPA (*Remotely Piloted Aircraft*), é uma aeronave que não necessita de pilotos humanos embarcados para ser operada. Este tipo de veículo pode ser comandado a

distância (tele-operados) ou por meios computacionais embarcados, funcionando com ou sem a intervenção humana, de maneira autônoma.

Também chamados de *drones* (zangão, em inglês), estes podem ser divididos, segundo a tabela 2.4, em tipos de asa fixa ou giratória. Os VANTs foram projetados originalmente para fins militares, mas atualmente possuem diversas aplicações como monitoramento de áreas, na cartografia, em missões de busca e resgate, na agricultura e no acesso à regiões críticas (GERCINA GONÇALVES DA SILVA; GONZAGA, 2015).

TAB. 2.4: Classificação de VANTs pelo tipo de asa

Classificação de VANTs	Indicação de uso
VANTs de asas fixas	São indicados para sobrevoar uma área extensa ao ar livres
VANTs de asas rotativas	São indicados para sobrevoar ambientes fechados e a baixa altitude

Segundo a ANAC (DE AVIAÇÃO CIVIL, 2015): "*Aeromodelos são as aeronaves não tripuladas remotamente pilotadas usadas para recreação e lazer e as aeronaves remotamente pilotadas (RPA) são as aeronaves não tripuladas utilizadas para outros fins como experimentais, comerciais ou institucionais.*"

Os dois tipos (aeromodelos e RPAS) só podem ser operados em áreas com no mínimo 30 metros horizontais de distância das pessoas não anuentes, ou não envolvidas com a operação, e cada piloto remoto só poderá operar um equipamento por vez (DE AVIAÇÃO CIVIL, 2016).



FIG. 2.5: VANT de asa rotativa do tipo quadricopter utilizado nesse trabalho

O DCEA (Departamento de Controle do Espaço Aéreo) é o órgão do Comando da Aeronáutica que tem por missão planejar, gerenciar e controlar as atividades relacionadas ao controle do espaço aéreo, à proteção ao voo, ao serviço de busca e salvamento e às telecomunicações do Comando da Aeronáutica. Através da portaria ICA-1040, regulamenta a operação de *drones* para fins civis. Essa portaria especifica os locais permitidos para voos, limites de operações e apresenta o SARPAS (Solicitação de Acesso de Aeronaves Remotamente Pilotadas). O SARPAS foi desenvolvido com o objetivo de facilitar a solicitação de acesso ao Espaço Aéreo para o uso de Sistemas de Aeronaves Remotamente Pilotadas (RPAS/DRONES) no Brasil.

Segundo o DECEA (DECEA, 2016), entende-se por aeronave totalmente autônoma quando não há a possibilidade de intervenção por parte do piloto. Nessa interpretação, as aeronaves remotamente pilotadas estarão sujeitas a utilização do espaço aéreo apenas com a devida atribuição de responsabilidades de um piloto. Segundo a nomenclatura desse órgão, a operação com base na utilização da navegação por *waypoints* é considerada Automática, podendo o piloto remoto assumir a qualquer momento o controle do veículo. Diferentemente do voo autônomo, operações de voo automáticos tem o seu acesso ao espaço aéreo autorizado, em conformidade com as legislações brasileiras.

Cabe ressaltar que por existir um operador capaz de controlar a aeronave a qualquer momento, todas missões executadas durante os experimentos são caracterizadas como automáticas, segundo o DECEA. Porém, em respeito à literatura acadêmica da área, foi optado por adotar o nome de voo autônomo nos textos desse trabalho.

## 2.9 TRABALHOS RELACIONADOS

Em Ananthanarayanan e Stoica (2009), os autores citam a tendência de dispositivos móveis embarcarem cada vez mais interfaces de rede com características complementares e que apesar da relação de banda de comunicação e consumo energético ser eficiente, o seu consumo em *idle* (espera) é alto, reduzindo a autonomia da bateria nos telefones. Os autores propõem o Blue-fi, uma solução desenvolvida para monitorar o ambiente através da interface bluetooth e da triangularização de torres GSM, habilitando as funções de WiFi apenas quando necessário (quando os sensores indicarem as proximidades de um ponto de acesso WiFi conhecido), proporcionando grande aumento na autonomia do dispositivo.

Outro alerta sobre a questão de eficiência energética em sistemas de localização é feito em (CHENG; HSIU, 2015), onde os autores criam um algoritmo de programação dinâmica para demonstrar uma otimização de energia através de agendamentos para o

consumo de informação pelos sensores, de modo que não prejudiquem a funcionalidade das aplicações que necessitem da estimativa de localização. Em (NEISHABOORI; HARRAS, 2013), para contornar o problema de eficiência energética em soluções de estimativa de localização de dispositivos *indoor*, os autores abordam a medição por *cluster* de alvos, se baseando em padrões de deslocamento e não no comportamento de um indivíduo.

Um estudo sobre privacidade de dispositivos bluetooth é feito em *Tracking of Mobile Devices through Bluetooth Contacts*, em que (RISTANOVIC et al., 2010) demonstra que ativos que mantêm a sua interface bluetooth habilitada podem ser rastreados por uma assinatura única e, com uma massa de dados regionais suficientemente grande, pode-se inferir onde o aparelho esteve em um certo período, por intermédio de coordenadas GPS de um sensor. (MAO et al., 2007) provê uma visão sobre técnicas de medição em localização de redes de sensores, mostrando como algoritmos de *one-hop* para cálculo de distâncias são mais populares, apresentando uma investigação sobre algoritmos de conexão *multi-hop* para estimativa de distâncias.

(HAEBERLEN et al., 2004) demonstra um sistema de localização por técnicas probabilísticas utilizando apenas um intensificador de sinais no padrão IEEE 802.11. Obtendo resultados *indoor* de 95% de acerto e com apenas 3 medições de intensidade de sinal. Haerberlen et al. (2004) ainda cita que diversas aplicações sensíveis à localização não precisam de precisões menores do que 2 metros, apresentando um modelo topológico simplificado em células para facilitar o mapeamento computacional diminuindo a resolução em troca de uma redução considerável no tempo de treinamento. O mesmo utiliza o modelo de sensores *Gaussian fit* que é mais robusto e requer menor tempo de treinamento comparado a modelos de sensores que utilizam histogramas completos de força de sinal.

Para localização do tipo *indoor*, múltiplos sensores são abordados na tese de Priyantha (2005), criando um estudo de caso onde os *beacons*, além dos *access points* WiFi, utilizam emissores de ultrassom para aumentar a precisão da estimativa de localização dentro de uma edificação. Esse uso misto de sensores possibilita uma precisão *indoor* de 10cm e apenas 3° em sua posição.

Em (WEBER et al., 2010), os autores dissertam sobre as limitações da estimativa de posicionamento utilizando WPS (*Wireless Positioning Systems*) e GPS. Apontam também que mesmo com técnicas de *map-matching*, pontos de acesso WiFi são imprecisos a priori, pois sua posição original já é uma estimativa. Apresentam o *selective look-ahead map-matching*, ou SLAMM, como técnica em tempo real de localização de trajetórias.

Herfurt e Mulliner (2004) sugerem uma técnica eficiente para determinar as propriedades de um dispositivo bluetooth para uma série de propósitos. Utilizando técnicas

similares ao IP *fingerprinting* utilizado em ferramentas como NMAP (NMAP.ORG, 2016), o *Blueprinting* pode determinar o fabricante, modelo do dispositivo e a versão do *firmware* de certos aparelhos através de consultas simples disparadas contra o receptor bluetooth.

O *Drone-sense* (YU et al., 2016) é uma proposta de coleta autônoma de sinais de redes WiFi em um ambiente 3D para otimizar o estudo sobre a propagação de ondas. Propiciando um método eficiente de coleta de dados para analisar modelos de propagação da cobertura de redes sem fio.

Jenks (2014) apresenta um design de fácil montagem para detecção de redes sem fio no padrão IEEE 802.11 utilizando ferramentas *open source* para *wardriving* e componentes de *hardware* de baixo custo e miniaturizados para a descoberta de *beacons* provisórios (*Rogue AP*). Embarca programas como Kismet e Airodump-NG, para a coleta de pacotes de radiofrequência.

Abramov et al. (2013) descreve a criação de um protótipo de veículo aéreo não tripulado para a validação de sistemas de segurança em TI (execução de *pentest*, em inglês). Segundo os autores, dispositivos desse tipo tem maior oportunidade de penetrar em perímetros restritos e atacar a infraestrutura de redes sem fio. Para o projeto são empregados apenas um VANT Parrot AR.Drone 2 e uma placa de computador Raspberry Pi (FOUNDATION, 2016b). A partir das redes sem fio vulneráveis detectadas, um algoritmo de ataque é empenhado para o cenário.

Em *Opportunities and Challenges in Crowdsourced Wardriving* (SAPIEZYNSKI et al., 2015), os autores mencionam as dificuldades da comunidade científica ao acesso à grandes bases de dados de localização e criam um protótipo para coleta massiva de *access points* por *crowdsourcing* a fim de montar uma base de dados e endereçar os desafios inerentes à sua administração bem como o seu nível de cobertura para obter precisão.

Uma proposta para definição de localização *indoor* sem o prévio conhecimento da topologia da estrutura e da posição dos *beacons* é discutido em (CHINTALAPUDI et al., 2010). O trabalho propõe o algoritmo *EZ Localization*. Sem depender de pré-calibração, esta ferramenta utiliza algoritmos genéticos para estimar a localização interna de uma edificação monitorando ensaios prévios de outros aparelhos que entraram anteriormente.

Borealis é uma ferramenta proposta em (ZHANG et al., 2011) que dispensa o uso de *hardwares* específicos para a captura de posições de pontos de acesso WiFi, utilizando *smartphones* normais equipados com sensores de uso cotidiano. A pesquisa aponta que apenas a rotação do emissor é suficiente para alterar a sensibilidade do sinal de radiofrequência recebido, demonstrando que todo aparelho pode ser utilizado em certo grau como uma antena direcional de transmissão e recepção WiFi. Essa técnica foi utilizada

com sucesso em ambientes externos para a detecção da direção de um ponto de acesso.

Em (KREVL; CIGLARIC, 2006), os autores descrevem as possibilidades de medição de posicionamento através de múltiplos sensores, como redes WiFi e GSM, *beacons* bluetooth e sistemas de Localização por satélite (GPS). Em adição, é proposto um *framework* para o desenvolvimento de aplicações distribuídas espacialmente.

Thomas King (2006) analisa diferentes abordagens de sistemas de localização e seus provedores, e oferece uma perspectiva sobre as tecnologias em desenvolvimento na área, em particular sobre as tecnologias de localização *indoor*. São considerados em sua abordagem algoritmos de fusão de sensores alegando que a sua precisão quanto a estimativa de posição de um emissor de radiofrequência poderá ser melhor estimada.

A tabela 2.5 exibe um comparativo entre os trabalhos relacionados na área de localização, fusão de sensores e aplicação de *drones*. Como analisado na seção 2.9, diversos artigos apresentam trabalhos de localização aplicando um sistema sensorial único ou por sensores de diferentes categorias, entre eles: sonares, unidades inerciais de medição e rádios. A utilização do padrão bluetooth nos trabalhos costuma se ater como um recurso extra para economia de energia ao sensor WiFi que por sua vez é utilizado como rádio principal nas estimativas de localização. Não foram encontradas aplicações de VANTs na localização de radiotransmissores em múltiplos padrões, contando apenas com o processamento embarcado e fusão de sensores, na bibliografia pesquisada.

TAB. 2.5: Comparação entre Trabalhos Relacionados

Referência	Técnicas Utilizadas	Observações
(MAO et al., 2007)	Bluetooth	demonstração de várias técnicas de localização
(CHENG; HSIU, 2015)	<i>wardriving</i>	propõe um sistema de agendamento de uso do sensor.
(KIM et al., 2006)	<i>wardriving</i>	Problemas de sinalização <i>indoor</i>
(ANANTHANARAYANAN; STOICA, 2009)	Bluetooth	utilização de múltiplos sensores para economia energética.
(RISTANOVIC et al., 2010)	Bluetooth	rastreamento de dispositivos bluetooth.
(YU et al., 2016)	<i>wardriving</i> e <i>drones</i>	medição de APs de forma automatizada
(THOMAS KING, 2006)	<i>sensor fusion</i>	uso de <i>sensor fusion</i> para estimativa de localização
(CRIVELLA, 2015)	<i>sensor fusion</i> e <i>drones</i>	uso de <i>sensor fusion</i> para otimizar a navegação de VANTs

### 3 METODOLOGIA

A determinação de uma técnica de localização em redes de sensores sem fio pode ser tomada a partir de uma das soluções apresentadas na figura 3.1. Dividido em 3 estratégias de posicionamento diferentes: Lateralização, Proximidade e Angulação - cada abordagem pode recorrer a diferentes técnicas de estimativa de distância que por sua vez dependem de sensores específicos para a estimativa. A abordagem desse trabalho dentro da árvore taxonômica pode caracterizá-lo como uma técnica de lateralização utilizando Força de Sinal (RSSI) por múltiplos sensores de RF (radiofrequência).

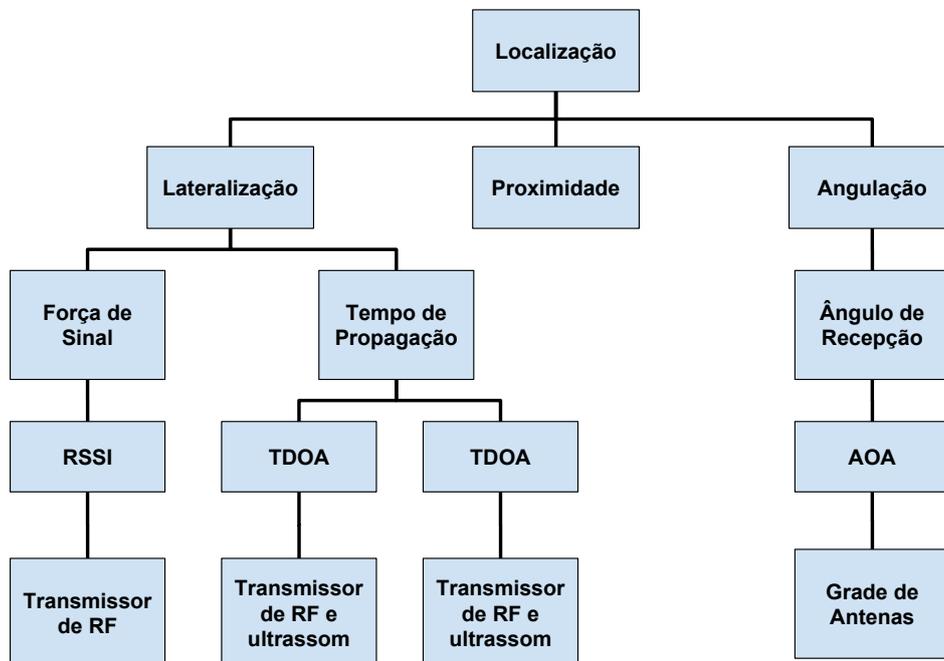


FIG. 3.1: Taxonomia de técnicas de localização em WSN.

Alinhado com os objetivos descritos na seção 1.3, esse trabalho demonstrará a aplicação da técnica de localização proposta por um sistema autônomo embarcado em veículos aéreos não tripulados. A determinação da posição de um alvo consistirá na estimativa de distância a partir de múltiplos pontos de captura de sinal durante um curso de voo. O resultado dessa estimativa de distância será otimizado com o uso de coletas em múltiplos padrões de transmissão de radiofrequência através de técnica de fusão de sensores. A determinação de posição do emissor será tomada através de uma técnica baseada em trilateração; de forma que, através de 3 ou mais estimativas de distância em pontos distintos,

se poderá inferir sobre a posição do alvo.

Durante o desenvolvimento da metodologia apresentada, foi realizado um estudo sobre as pesquisas relevantes na área de localização em WSN e sobre as tecnologias adotadas. Após a fase de mapeamento de dependências, foi criado um *framework* que serviu como ponto de partida para a modelagem do algoritmo proposto. O desenvolvimento de *software* ocorreu em paralelo à montagem do computador de bordo da plataforma de voo. A montagem do VANT para a coleta e estimativa de sinais foi personalizada de modo que a ferramenta de detecção, o sistema computacional e os sensores fossem possíveis de serem embarcados no veículo de forma viável à proposta. Informações sobre essa montagem podem ser consultadas em detalhes no anexo 8.3. Sensores dedicados à coleta de sinais de radiofrequência foram configurados no sistema embarcado e ferramentas de apoio à operação aérea foram selecionadas para a criação de missões de voo e monitoramento dos ensaios.

### 3.1 DETERMINAÇÃO DA TÉCNICA DE LOCALIZAÇÃO

Técnicas de localização baseadas em análise de RSSI (*Received Signal Strength Information*) costumam ser implementadas com baixa complexidade e não necessitam de sincronização de relógio entre os nós âncora e o nó sensor. Sua operação também tende a ser de baixo consumo de recursos computacionais e podem ser utilizados *hardwares* não dedicados, como placas de redes WiFi. Apesar de não necessitar de mudanças no protocolo de rede, flutuações inerentes à propagação de ondas de radiofrequência no meio acarretam em uma baixa precisão em suas estimativas.

Técnicas de ToA (*Time of Arrival*) e TDoA (*Time Difference of Arrival*) são consideravelmente precisas, sendo muito recomendadas em soluções de localização dentro de ambientes *indoor*. Porém, o seu uso depende do sincronismo do relógio entre emissor e receptor e de um *hardware* dedicado, ainda não presente em *smartphones* e outros bens de consumo populares. Por ser uma solução de propósito específico, a sua implementação costuma ser de maior complexidade, necessitando muitas vezes de maior consumo elétrico.

Técnicas de estimativa por ângulo de chegada (AoA) podem operar como uma espécie de radar e a sua implementação pode utilizar *hardwares* de propósito geral montado em uma estrutura direcional com isolamento de sinais. Seu uso geralmente conta com mais de um receptor e um mecanismo de rotação sobre um eixo de modo a analisar a distância de um emissor ao mesmo tempo que estima a sua posição por angulação. Apesar de possibilitar a detecção de posição de um alvo sem precisar deslocar o nó sensor, essa

técnica necessita de uma estrutura apropriada e, em certos casos, aumenta o consumo. Sua implementação também é consideravelmente mais complexa.

A partir da árvore taxonômica da figura 3.1, foi criada uma tabela comparativa entre diferentes padrões de estimativa de distanciamento (vide tabela 3.1). A partir de cada técnica analisada na literatura, foram verificados suas vantagens e limitações, bem como a necessidade de componentes tecnológicos específicos para a sua utilização prática. Através dessa análise, e pelo desejo de utilizar padrões de ampla utilização, de propósito geral e, principalmente, acessíveis, foi adotado o padrão por força de sinal (RSSI) para determinação de distância e a técnica de lateração para estimativa de posição de um alvo.

TAB. 3.1: Comparação entre técnicas de estimativa de distância em WSN

	Precisão	Sincronia	Hardware Dedicado?	Antena	Observações
RSSI	baixa	não	não	não	pode ser implementado sem nenhuma infraestrutura dedicada
ToA e TDoA	alta	sim	sim	não	necessita de protocolo ou hardware para a finalidade. Maior Complexidade
AoA	alta	não	não	sim	permite que o sensor fique estático

Restrições de recursos na aeronave, como alimentação, espaço para instalação de componentes eletrônicos e barramento de comunicação com o computador de bordo, também foram fatores decisivos na escolha da técnica.

### 3.2 ARQUITETURA PROPOSTA

A arquitetura proposta (apresentada na dissertação), conta com 4 componentes principais: 1) O *framework* de operação, que direciona a execução da metodologia; 2) a modelagem do algoritmo, e por sua vez do *software* embarcado, de maneira que atenda aos componentes do *framework*; 3) a definição da plataforma de voo, respeitando às necessidades de coleta e modos de operação pretendidos; e 4) plataforma computacional embarcada, que conta com os recursos necessários para execução do *software* desenvolvido, controle da aeronave e interface com os sensores utilizados dentro dos limites de operação.

### 3.2.1 *FRAMEWORK* METODOLÓGICO

O relacionamento entre os componentes do *framework* proposto é descrito na figura 3.2: A primeira etapa consiste na coleta de dados utilizando rádios de diferentes padrões. O resultado de cada amostra é registrado em base com as coordenadas de voo para depois ser consumido pelas ferramentas de cálculo de distância. Essas estimativas, específicas para cada tecnologia, também são registradas com as mesmas coordenadas das amostras, como relacionamento principal, para então serem utilizadas pelo algoritmo de fusão de sensores. A ferramenta de trilateração, com base nas distâncias aferidas a cada instante de coleta, oferece um resultado final de posicionamento relativo à aeronave. Ao final, coordenadas absolutas de voo, obtidas em paralelo pelo sistema de GNSS, também podem ser utilizadas para gerar uma saída com o geoposicionamento da fonte emissora através de bibliotecas de trilateração, resultando em uma estimativa de posição do alvo.

Funcionando como ferramenta de *wardriving*, os componentes responsáveis pela captura de dados de redes WiFi, Bluetooth e Bluetooth Low Energy são as bibliotecas IEEE8211Lib, BTLib e BLELib, respectivamente. Toda informação obtida é persistida em arquivos de texto ou em base SQL, contendo uma sequência histórica dos registros de voo. A modelagem do banco tem foco na retenção das informações geradas pelas bibliotecas e na manutenção do sincronismo entre os dados capturados, as localizações da aeronave e as estimativas do posicionamento e distância do alvo.

Os valores capturados e armazenados são usados nos cálculos dos algoritmos de localização, tais como: valor do RSSI, nome do dispositivo em *broadcasting* ou *advertisement* (no caso de pacotes bluetooth), qualidade do sinal e frequência de transmissão. Para a determinação da identidade de uma fonte são utilizados o endereço físico do *hardware* (*MAC Address*), tipo de criptografia adotada, nome da rede e modo de interface. Porém, outros atributos podem ser configurados para garantir segurança e maior precisão na associação com o alvo.

As bibliotecas de captura e a biblioteca que implementa o algoritmo de fusão de sensores foram desenvolvidas em Python. Dessa forma, o *framework* será modularizado por bibliotecas, responsáveis por cada método utilizado, proporcionando uma arquitetura de baixo acoplamento e escalar, para futuramente suportar novas tecnologias, caso necessário. Essa linguagem também facilita a conexão com o sistema de controle da aeronave pelas bibliotecas DRONEKIT (ROBOTICS, 2016b).

O algoritmo executa de forma interativa com a rota de voo registrada na aeronave e o disparo das bibliotecas de *wardriving* é associado com os pontos de interesse de uma

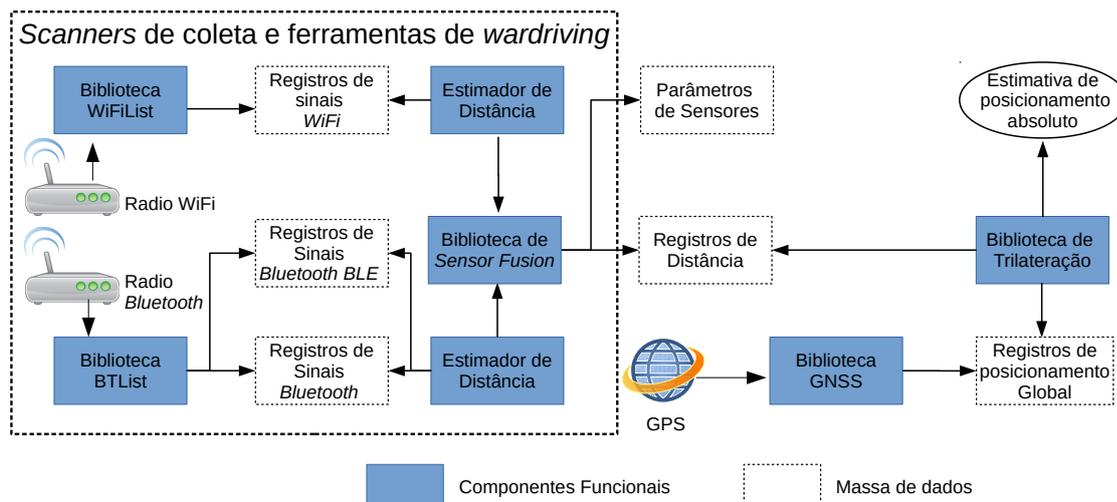


FIG. 3.2: *Framework* proposto para a metodologia de localização

missão de voo preestabelecida. Essa dependência garante que a coleta de amostras seja realizada com estabilidade. A estimativa de distância pelas bibliotecas WiFi e bluetooth, bem como a fusão de sensores e determinação de posicionamento pela ferramenta de trilateração, trabalham com os dados já capturados e não necessitam que a aeronave esteja parada. A partir do 3º ponto de coleta, a biblioteca de trilateração é ativada, lendo todas as estimativas de distância calculadas em cada coordenada, retornando uma estimativa de localização do alvo emissor. A ordem de execução do algoritmo projetado por ser acompanhada no Código 1.

---

**Código 1:** Ordem de execução do algoritmo de localização

---

**Data:** *Coordenadas e Distancias*

**Data:** *PontosdeInteresse*

**Result:** *Estimativadeposicionamento*

```
1 Carga Inicial;
2 while voo Autônomo do
3   lê Ponto Atual;
4   if Ponto Atual = PontosdeInteresse then
5     coletar sinais WiFi;
6     coletar sinais Bluetooth;
7     executar fusão de sensores;
8     calcular distância;
9     salvar resultados;
10  else
11    Aguarda;
12  end
13  if Ponto Atual Count >= 3 then
14    executar trilateração;
15    salvar resultados;
16    exibir EstimativadePosicionamento;
17  end
18 end
```

---

### 3.2.2 ESTIMATIVA DE DISTÂNCIA

O problema de localização envolve dois estágios: estimativas de distância e lateração. O cálculo de distância, baseado em RSSI na metodologia proposta, utiliza a técnica log-normal de atenuação por sombra (*Lognormal Shadow Model*) e depende previamente da calibração dos parâmetros ambientais e do RSSI médio, registrado a uma distância conhecida. Uma vez parametrizada, por essa técnica se calcula o decaimento da força de sinal recebido com um aumento da distância entre receptor e emissor. A força de sinal recebida (variável **RSS**) pode ser expressada de acordo com a equação 3.1.

$$\mathbf{RSS} = \mathbf{P}_0 - 10\alpha \log \mathbf{d} \quad (3.1)$$

A partir dela, a distância **d** da posição de um nó sensor até o emissor pode ser

determinada pela seguinte equação (3.2):

$$\mathbf{d} = 10 \left( \frac{\mathbf{P}_0 - \text{RSS}}{10\alpha} \right) \quad (3.2)$$

Onde  $\mathbf{P}_0$  é a potência recebida pelo sensor a uma distância de 1 metro,  $\mathbf{d}$  é a distância entre a aeronave e o alvo; e  $\alpha$  é o componente ambiental de *path loss*.

O algoritmo implementado (ver algoritmo 2) deve executar uma leitura baseada no número de amostras desejadas. A ferramenta de captura de sinais WiFiList previamente registra uma lista de todos os *scans* efetuados no ambiente baseado nessa repetição. Em posse desse registro, o algoritmo de estimativa de distância faz a leitura apenas das entradas relacionadas ao dispositivo alvo baseado em atributos de identificação como o seu *Mac Address*, por exemplo. Ao registrar uma matriz apenas com as entradas desejadas o algoritmo faz a leitura do RSSI de cada coleta e retorna uma distância estimada para cada amostra. Por fim, é feita uma média de distâncias e esse valor é retornado. Na linha 7 desse algoritmo, observa-se que amostras com valores 0 ou *Null* são descartados, uma vez que em ambientes reais não são encontrados esses resultados e geralmente indicam falha no registro de captura feito pelo módulo WiFi do Sistema Operacional.

---

**Distancia 2:** Execução do algoritmo de estimativa de distância

---

**Data:** *Coordenadas*, *Count*, *WiFiListoutput* e *numerodeamostras*

**Data:** *MacAddress*

**Result:** Estimativa de Distancia

```
1 while count ≤ numerodeamostras do
2   lê WiFiList output;
3   for i ∈ WiFiListoutput do
4     i ← Identification;
5     if i = MacAddress then
6       i ← RSSI;
7       if i ≠ 0 then
8         count = count + 1;
9         executa lognormalshadowfunction;
10        SUMDistance ← Distance;
11        executa AVGDistance;
12      end
13    end
14  end
15  retorna AVGDistance;
16 end
```

---

### 3.2.3 ESTIMATIVA DE POSIÇÃO

Mesmo que o problema de trilateração possa ser resolvido com uma técnica geométrica, essa abordagem não é prática em sistemas computacionais embarcados, sendo necessários modelos matemáticos precisos para a sua resolução. Porém, se os círculos construídos pelas distâncias estimadas se cruzarem em múltiplos pontos ou em nenhum, a solução estará comprometida (vide figura 3.3).

A resolução da trilateração pode ser tratada do ponto de vista de otimização. Dessa maneira, busca-se encontrar o ponto  $\mathbf{X} = (\rho_{\mathbf{x}}, \phi_{\mathbf{x}})$  que ofereça a melhor aproximação da posição real  $\mathbf{P}$ .

Calcula-se a distância  $\mathbf{L}_i$  entre cada ponto de coleta de amostras executado pela aeronave. Se essas distâncias se encontrarem perfeitamente em um ponto entre os comprimentos  $\mathbf{d}_i$ , então o  $\mathbf{X}$  estimado de fato será o ponto  $\mathbf{P}$ . O quanto mais  $\mathbf{X}$  se afasta dessas distâncias, mais ele precisa ser assumido a partir de  $\mathbf{P}$ . Busca-se então encontrar

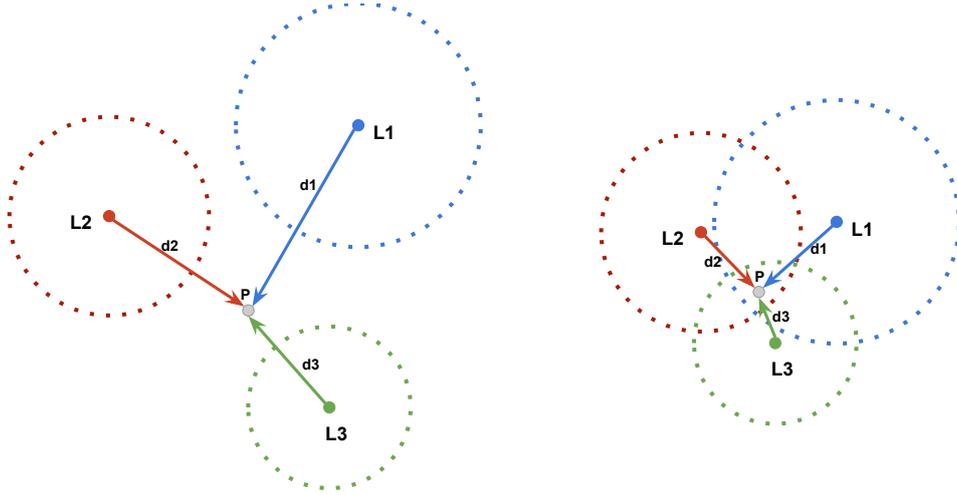


FIG. 3.3: Tolerâncias a pontos de estimativa fora da interseção dos círculos raio  $d_i$

$X$  que minimize uma determinada função de erro. Para um  $X$ , existirão 3 funções de erro - uma para cada ponto de coleta, que funcionará como nó âncora para o sistema de estimativa de posição (equação 3.3).

$$\begin{aligned}
 e_1 &= d_1 - \text{dist}(\mathbf{X}, L_1) \\
 e_2 &= d_2 - \text{dist}(\mathbf{X}, L_2) \\
 e_3 &= d_3 - \text{dist}(\mathbf{X}, L_3)
 \end{aligned} \tag{3.3}$$

Para considerar a contribuição de cada uma dessas fontes de erro na resolução, utilizamos a média de seus quadrados, evitando assim a entrada de valores de erro positivos e negativos no sistema, que poderiam se cancelar. O resultado obtido é o erro quadrático médio (*mean square error*, MSE)

$$\frac{\sum [d_i - \text{dist}(X, L_i)]^2}{N}. \tag{3.4}$$

A vantagem dessa solução é a possibilidade de considerar um número arbitrário de pontos na resolução, desde que se use no mínimo 3 em diferentes locais de coleta. Para achar o ponto  $\mathbf{X}$ , que minimiza o erro quadrático médio, utiliza-se a técnica (*Limited-memory Broyden-Fletcher-Goldfarb-Shanno*), uma variação de maior eficiência computacional do BFGS, um algoritmo da família de métodos *quasi-Newton*. Sua utilização será provida pela biblioteca *opensource* Scipy da linguagem Python.

A partir do terceiro ponto de coleta de amostras, o algoritmo principal chama o método de cálculo de trilateração repassando para esse algoritmo uma lista de distâncias estimadas e as coordenadas onde as amostras foram coletadas (ver algoritmo 3). O algo-

ritmo de estimativas de posição por sua vez ordena o *array* de distâncias e converte cada coordenada em radianos (linha 4) para os cálculos de circunferência, com o raio igual a distância, entre os pontos de coletas (linha 8). Um algoritmo de otimização recursivo é disparado na linha 9 e retorna o centroide entre todas as circunferências calculadas, de todos os pontos de coleta até o momento de sua execução.

---

**Posicao 3:** Execução do algoritmo de estimativa de posição

---

**Data:** *Coordinates, Count e Distances*

**Data:** *RaiodaTerra, RadCoordinates*

**Result:** *EstimativadePosicao*

```

1 for  $i \in \text{Coordinates}$  do
2   | sort Distances;
3   | for  $i \in \text{Coordinates}$  do
4     |   executa Conversão em Radianos;
5     |    $\text{RadCoordinates} \leftarrow \text{ConversaoemRadianos}$ ;
6     | end
7     | for  $i \in \text{RadCoordinates}$  do
8       |   executa Calculo de distancia entre círculos  $i$  e  $i + 1$ ;
9       |   executa método de calculo MSE;
10      |   executa método de otimizacao L-BFGS-B;
11      | end
12     | retorna EstimativadePosicao;
13 end

```

---

### 3.2.4 FUSÃO DE SENSORES

Dado um transmissor de radiofrequência em múltiplos protocolos, determinar a sua localização de origem baseado nas amostras coletadas por diferentes sensores (WiFi (W) e *bluetooth* (B)) em diferentes instantes, por um mesmo agente.

Os sensores adotados são capazes de determinar a distância relativa da fonte emissora, em função da intensidade de sinal  $\mathbf{I}$  capturado e o posicionamento no instante da coleta de cada amostra  $\delta$ . Registrando uma série temporal, cada amostra  $\delta_i$  é composta por uma tupla de valores, que conta com a localização da aeronave  $\mathbf{P}_i$  no instante  $\mathbf{t}_i$  e as informações de Intensidade de sinal  $\mathbf{I}_i^{\mathbf{W}}$  e  $\mathbf{I}_i^{\mathbf{B}}$ , medidas de cada rádio bem como informações do respectivo erro associado  $\mathbf{e}_i^{\mathbf{W}}$  e  $\mathbf{e}_i^{\mathbf{B}}$ , conforme equação 3.5.

$$\langle P_i, t_i, I_i^W, e_i^W, I_i^B, e_i^B \rangle_{\delta_i} \quad (3.5)$$

Devido a características físicas de emissão de sinais, conclui-se que erros de estimativa desses são diretamente proporcionais à distância entre o ponto (P) e a origem de transmissão de radiofrequência. Para otimizar a solução de localização, espera-se a minimização dos erros e referentes aos dados de cada sensor, ou seja

$$\min e_i^j, \text{ onde } j \in \{WiFi, bluetooth\}. \quad (3.6)$$

Para tratar diferentes fontes de dados armazenadas em uma série temporal, o algoritmo de fusão de sensores adotado será o Filtro de Kalman.

### 3.3 VEÍCULO AÉREO NÃO TRIPULADO

Durante o desenvolvimento da pesquisa, inicialmente foi cogitada uma plataforma para voos experimentais do tipo asa fixa. Essa decisão inicial se deu em virtude de um veículo de mesma natureza já ser utilizado em pesquisas pelo laboratório e pelo conhecimento consolidado em sua operação. Porém, essa solução mostrou-se inadequada devido a natureza do voo pretendida durante a fase experimental. Optou-se por um VANT do tipo asa rotatória de pequeno porte (ilustrado na figura 3.4) capaz de comportar a carga necessária pelas eventuais customizações, bem como controles precisos de altitude e velocidade, capacidade de decolar e pousar na vertical e pairar.



FIG. 3.4: VANT utilizado nos testes de voo, com carga paga de até 1,2 kg.

Baixo orçamento e facilidade de busca por componentes disponíveis no mercado também foram fatores considerados relevantes para o projeto. Uma lista detalhada com os componentes adotados, fornecedores e os seus custos pode ser consultada no apêndice 7.1.

O aeromodelo adotado é do tipo quadricóptero, utiliza a estrutura, motores e propulsores do fabricante DJI. Possui um *frame* principal, modelo F450, de 450mm de envergadura (diâmetro) e 282g sem os acessórios e um compartimento para eletrônica capaz de suportar até 1200g de carga extra com os rotores adotados. Suporta baterias LiPo do tipo 3S e 4S (5200mAh) e furações extras para customizações necessárias e para fixação do gabinete com o sistema computacional embarcado. Os motores adotados são de 960kv, modelo 2312 que, combinados com hélices de 9,4 polegadas, são capazes de erguer de 350g a 400g (cada um dos 4 motores) utilizando apenas 50% de sua capacidade (potência de *thrust*). O circuito de potência (ESC, *Electronic Speed Control*) que alimenta cada motor é do modelo E310, também da fabricante DJI. Um passo-a-passo detalhado sobre o processo de construção da aeronave pode ser visto no anexo 8.3.

### 3.3.1 SISTEMA EMBARCADO

O sistema computacional embarcado é composto por uma unidade de computação de bordo, capaz de atender às necessidades de execução do *software* desenvolvido dentro das restrições de operação; e uma unidade inercial de movimento que atende as funções necessárias do sistema de piloto automático: capaz de operar a aeronave através de seus componentes dedicados.

#### 3.3.1.1 COMPUTADOR DE BORDO

Como computador de bordo, foi adotado a placa Raspberry Pi 3 Model B (revisão 1.2). Esse modelo foi escolhido por incorporar um *chip* Broadcom BCM43438 que provê rádios para redes sem fio nos padrões IEEE 802.11n (2.4GHz) e *bluetooth* 4.1 (com suporte a *Bluetooth Low Energy*), dispensando o uso de periféricos no Barramento Serial Universal (*Universal Serial Bus*, USB). Essa placa tem 85,6mm de comprimento por 56mm de largura e 21mm de espessura, consumindo 1000mA/h, no máximo. Sua CPU possui 4 núcleos a 1,2 GHz (ARMv8 64-bits), além de suportar cartões de memória microSD e possuir uma interface de propósito geral (*General Purpose Input/Output*- GPIO) Foundation (2016a). Seu poder computacional alcança 462,07 MFLOPS, sendo que cerca de 10% é usado pelo *software* da controladora de voo.

Dentre os sistemas operacionais compatíveis com o hardware, foi adotado o Raspbian, um sistema Linux-ARM baseado na distribuição Debian. Sua escolha se deve ao grande suporte da comunidade de usuários e por já possuir módulos compilados em seu repositório para integração com os sistemas de voo *open source* adotados no projeto.

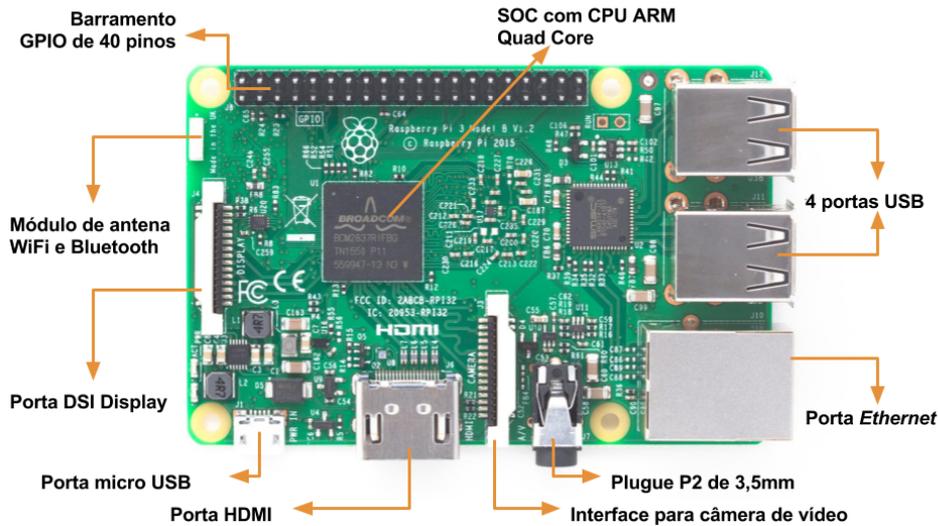


FIG. 3.5: Placa Raspberry Pi 3 Modelo B com o detalhamento de todas as suas interfaces

### 3.3.1.2 UNIDADE DE MEDIDAS INERCIAIS (IMU)

Para o controle de voo, foi adotada a placa Navio2 da fabricante EMLID. Esse hardware possui duas Unidades de Medidas Inerciais (*Inertial Measurement Unit* - IMUs) com acelerômetro, giroscópio e magnetômetro de três eixos, barômetro e GPS. Suporta Modulação por Largura e por Posição do Pulso (*Pulse Width Modulation* – PWM e *Pulse-Position Modulation* - PPM), e extensão de comunicação serial (*Universal Asynchronous Receiver/Transmitter* - UART). Suporta os padrões de barramento *Inter-Integrated Circuit* (I2C) e *Analog-to-Digital Converter* (ADC). A placa tem dimensões de 55mm de comprimento por 65mm de largura e peso de 23g (EMLID, 2016).

O processo de acoplamento do *hardware* de piloto automático com a placa-mãe representa a integração completa do sistema computacional embarcado. A configuração desse agente de controle exige a instalação de espaçadores no lado superior do Raspberry Pi, parafusos e cabeçalho de extensão da porta GPIO entre as placas, bem como a confecção de um gabinete personalizado para embarcar a solução na aeronave.

Sua escolha se deve ao suporte ao serviço de comando de voo Ardupilot e a possibilidade de integração da placa Raspberry Pi como um *Companion Computer* que ofereça poder de processamento e armazenamento necessários para o projeto e customizável sob demanda durante os ensaios. Um *Companion Computer* é um dispositivo integrado que pode navegar dentro de um veículo e controlar ou se comunicar com o piloto automático sob *links* de baixa latência. Aplicativos rodando nesse tipo de recurso computacional podem executar tarefas mais intensas com mais performance do que o *hardware* disponibilizado pelo sistema de piloto automático sozinho (ROBOTICS, 2016c).

### 3.3.1.3 SENSORES

Além dos diversos sensores de voo já embarcados na placa NAVIO2 (como explicado na seção 3.3.1.2), por suas portas USB, o Raspberry Pi possibilita a inclusão de mais periféricos ao sistema computacional de bordo, expandindo as suas capacidades.

Para a execução do *framework* proposto, foram adotados os seguintes rádios de comunicação como sensores dedicados ao sistema de localização: rádio no padrão IEEE 802.11 AC (2.4 e 5 GHz), fabricado por TP-Link (modelo Archer T2UH com *chipset* MediaTek MT7610U), placa com interface USB e antena omnidirecional de ganho de 2dBm; rádio no padrão bluetooth 4.0 (2.4 GHz), do fabricante Broadcom (modelo BCM20702A0), com suporte a BLE e antena omnidirecional impressa no próprio PCB.

A lista completa de todos os radiotransmissores envolvidos na aeronave, com sua interface de comunicação e frequências de operação, pode ser consultada na tabela 3.2

TAB. 3.2: Radiotransmissores embarcados no VANT

Modelo do Rádio	Freq. (Ghz)	Porta	Características
WiFi N Broadcom BCM43438	2.4	onboard	comunicação SSH/UDP
WiFi AC TP-Link Archer T2UH	2.4 e 5	USB	dedicado à localização
Bluetooth BCM43438	N/A	onboard	desativado pelo Kernel
Bluetooth BCM20702A0	2.4	USB	dedicado à localização
3DR SiK Radio	433 MHz	UART	telemetria
FlySky FS-IA6B (8ch)	2.4	canal PPM	operação manual

Além dos rádios embarcados, pode-se destacar sensores de navegação ligados a outros barramentos que expandiram o controle e estabilidade de voo do VANT empregado na fase experimental.

- a) Maxbotix Sonar EZ0 (I2CXL): dispositivo de ultra-som que opera como sensor de proximidade em distâncias entre 20cm e 4m para controle de elevação;
- b) Lidar Lite V3 Garmin: sensor laser com sensibilidade entre 20cm e 40m com alta taxa de frequência e maior precisão para manutenção de altitude da aeronave;

Foi adotada uma câmera para expandir a capacidade da plataforma e auxiliar na aferição de posicionamento do veículo. Devido a maior compatibilidade com o computador de bordo, foi adotado o modelo Raspberry Camera V2. Esse dispositivo possui um sensor de 8MP e oferece os modos de vídeo na resolução de 1080p, 720p ou VGA (a 30fps, 60fps e a 90fps, respectivamente). Para estabilizar a câmera, um *gimbal* de dois eixos (rolagem

e arfagem) foi construído com dois motores servo ligados diretamente à placa Navio2, tomando partido de suas IMUs para controle e estabilização.

A especificação completa de todos os sensores e recursos computacionais adotados nesse trabalho podem ser consultados no Anexo 8.4 - esse documento apresenta características de operação, fabricante e detalhes técnicos de cada hardware empregado.

#### 3.3.1.4 ADAPTAÇÕES

Para acomodação dos componentes escolhidos e para maior estabilidade no processo de pouso e decolagem, diversas peças plásticas em material ABS e PLA foram construídas. Tais peças foram projetadas de maneira personalizada para a aeronave e concebidas segundo o projeto descrito por Ricardo M. Bernardo e Rosa (2017). Foram utilizadas impressoras 3D em sua manufatura para que fossem rapidamente prototipadas e reproduzidas em caso de manutenção. Esses componentes foram usados principalmente para construção de cascos de proteção da eletrônica a bordo e do sistema de pouso, mas não são de uso obrigatório.

Devido a interferências eletromagnéticas detectadas nos primeiros voos de teste com VANT, foram desenvolvidas blindagens sobre componentes de navegação mais sensíveis, especialmente para bússolas digitais e antenas de GPS. Esses *hardwares* foram eleitos críticos uma vez que o seu mal funcionamento pode comprometer a performance do voo, causando acidentes.

Foram criadas especialmente para o veículo proteções de alumínio com 1mm de espessura no casco inferior da carapaça que abriga o computador de bordo e um disco de 2mm de espessura e 80mm de diâmetro na base da antena de GPS, todos ligados ao polo negativo da bateria. Essas proteções visam impedir principalmente os ruídos oriundos de circuitos de potência e motores posicionados mais abaixo do plano horizontal desses dispositivos.

Outros sensores, como rádios WiFi e bluetooth, com antenas omnidirecionais, também são sensíveis a interferências eletromagnéticas, mas não foram protegidos para estudar o impacto do *hardware* de um veículo de asa rotativa em sua operação normal.

#### 3.3.2 SISTEMA DE NAVEGAÇÃO

De modo que a aplicação fosse capaz de controlar o navegador automático durante uma missão de voo, foi utilizado o *framework* DroneKit. As suas bibliotecas possuem pacotes compilados para diversas distribuições de Linux que suportam Python. Toda a comunica-

ção entre a aplicação Python rodando no Raspberry Pi é feita pelo serviço Ardupilot que por sua vez comanda a placa Navio2 por instruções formatadas em protocolo MAVLink.

O serviço Ardupilot possui recursos para operar subsistemas de piloto automático para veículo de diferentes naturezas: tanto na versão Plane (para aeronaves de asa fixa), Copter (para aeronaves de asa rotativa) e Rover (para veículos terrestres). Configurações menos usuais, como antenas de rastreamento e múltiplas instâncias de veículos operando em enxames, também são suportadas.

O serviço é executado pelo S.O. GNU Linux e carregado durante o *Boot*. Por se tratar de um sistema de operação crítica para a aeronave, após inicializado, o Ardupilot é carregado no *firmware* da placa de navegação automática, podendo operá-la com baixa latência e alta disponibilidade. Alterações nas configurações do veículo feitos na camada de sistema operacional são aplicadas ao subsistema na NAVIO2 a cada vez que o serviço é reiniciado no Raspbian.

O Ardupilot possui um conjunto de modos de operação do veículo que definem comportamentos úteis para diferentes cenários de navegação. Dentre os mais utilizados no projeto, pode-se destacar:

- a) modo Stabilize: Ativa o modo de voo manual, onde o operador através de um rádio controle ligado à Navio2 tem controle total do veículo;
- b) modo Guided: Ativa o modo de voo assistido, onde só é possível operar o veículo por comandos MAVLink enviados ao Ardupilot;
- c) modo Alt\_Hold: Mantém o atributo altitude do veículo fixo, liberando ao operador apenas o controle dos movimentos de *Yaw*, *Pitch* e *Roll*;
- d) modo Loiter: Mantém os atributos altitude e orientação do veículo fixo, liberando ao operador apenas o controle dos movimentos de *Pitch* e *Roll*;
- e) modo Land: Ativa o modo de aterrissagem automática (sem a necessidade de intervenção humana no processo) no ponto atual onde a aeronave se encontra;
- f) modo RTS (*Return do Land*): Ativa o modo de aterrissagem automática no ponto onde a aeronave decolou, seguindo o menor caminho a uma altitude pré-configurada;
- g) modo AUTO: Ativa o sistema de navegação autônoma. O piloto automático carrega uma missão de voo preestabelecida e inicia a navegação.

### 3.3.3 ESTAÇÃO DE CONTROLE EM SOLO

O serviço Ardupilot suporta comandos enviados por entidades externas através do padrão MAVLink. Esses comandos permitem que outros programas sejam capazes de operar um veículo não tripulado dentro das limitações do protocolo. Dessa maneira Estações de Controle em Solo, ou *Ground Control Stations* (GCS), podem ser empregadas como interface de comando e controle de um veículo, apresentado uma interface mais amigável e funcional para o operador.

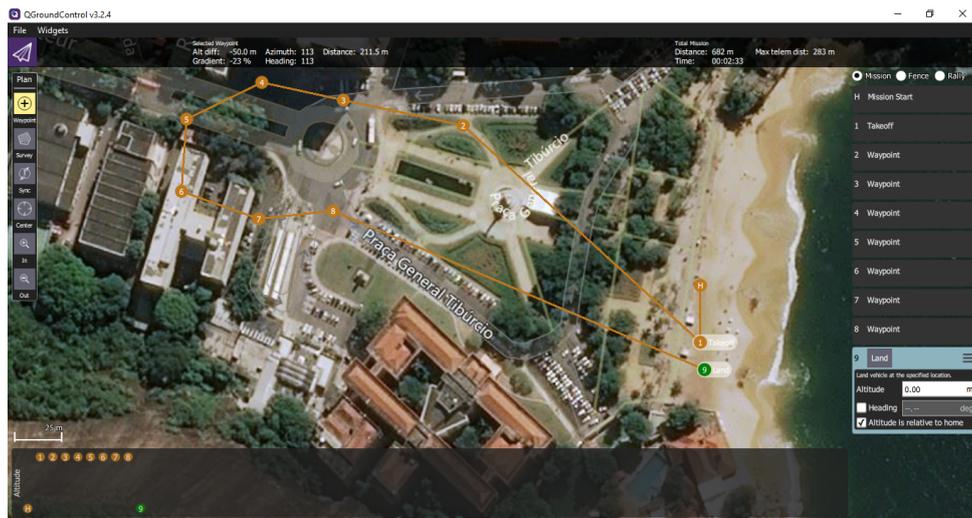


FIG. 3.6: Tela de edição de missões de voo pelo GCS QGroundControl

As estações de controle em solo também funcionam como ferramenta de edição de missões de voo, oferecendo uma interface simplificada para a transposição de pontos sob uma camada de mapa planejada no padrão WGS85. Durante a edição de uma missão de voo, cada ponto é definido segundo a expectativa de operação da aeronave em modo automático (AUTO). Dentre as opções, foram utilizados os seguintes tipos de pontos de missão na fase experimental desse trabalho:

- a) Waypoint: Define as coordenadas de latitude, longitude e altura de destino do VANT. Pode-se determinar velocidade e orientação do veículo a cada ponto;
- b) Land: Define o processo de pouso do veículo, a altura da aterrissagem e local;
- c) Takeoff: Define o processo de decolagem do veículo, sua altura final e local;
- d) Hold\_position: Define um ponto de pausa do *drone*. São definidas as coordenadas de latitude, longitude e altura bem como o tempo e posição de espera.

Após a edição, a missão de voo é salva em arquivo texto, em formato CSV ou JSON (em GCS mais modernos). Para carregar no *firmware* da aeronave, a instância da entidade da aeronave recebe um *upload* da missão por comandos MAVLink. O VANT só pode receber uma missão por vez. Com a sua carga em memória, a missão será iniciada imediatamente ao alterar o modo de operação da aeronave para o modo AUTO.

O GCS monitora a evolução da aeronave em mapa, recebendo atualizações de suas coordenadas e logs de mensagens (no formato TLog), a cada instante através de um canal de comunicação TCP, UDP ou Serial (para utilização de radiotransmissores com maior resiliência, como o 3DRobotics 433MHz). As estações de comando em solo servem uma interface mais completa do estado da aeronave e da missão de voo e podem ser executadas a partir de um *workstation*, celular ou *tablet*.

Durante um ensaio de voo, mais de uma aplicação de GCS pode ser empregada para o monitoramento do VANT, por possuir funcionalidades particulares que atendem melhor a cada cenário. As aplicações utilizadas durante os testes de voo foram:

- a) Mission Planner: para execução em *Workstation*, em ambiente Microsoft Windows e auditoria de logs de voo
- b) QGround Control: Para execução em *tablet*, em ambiente Android, como monitor auxiliar ao operador do *drone*;
- c) APM Planner 2: para execução em *workstation*, em ambiente GNU Linux, edição de missões de voo e auditoria de logs de voo

### 3.4 DESENVOLVIMENTO EM *SOFTWARE*

Durante a implementação da metodologia, todas as bibliotecas de captura, e as bibliotecas que implementam o algoritmo de fusão de sensores e trilateração, foram desenvolvidas em Python. Dessa forma, o *framework* foi modularizado em pacotes, cada um responsável por um conjunto de métodos afins, proporcionando uma arquitetura de baixo acoplamento e escalar, para futuramente suportar novas tecnologias. Essa linguagem também facilita a conexão com o sistema de controle da aeronave pelas bibliotecas DRONEKIT (ROBOTICS, 2016b).

Para atender diferentes cenários, foram desenvolvidos 3 Programas Principais, cada qual com uma finalidade específica dependendo do caso de uso. As bibliotecas de apoio retêm os métodos necessários que são utilizados por cada programa para provar a metodologia proposta:

- a) **mainFly**: Programa principal responsável por controlar a aeronave ponto-a-ponto e disparar as coletas dos sensores de maneira centralizada e arbitrária. Por manipular diretamente o sistema de voo da aeronave, as missões de voo preestabelecidas e a captura de amostras devem ser bem planejadas (preferencialmente testadas em um simulador) a fim de evitar acidentes.
- b) **mainMission**: Programa principal que observa uma missão de voo comandada por entidade externa e dispara a coleta de sensores em pontos (*waypoints*) pré-determinados sem interferir na rotina voo em execução. Essa solução é menos intrusiva e mais segura por não precisar controlar todas as variáveis inerentes à operação da aeronave. Porém, não administra o tempo de coleta e é dependente de uma missão de voo bem planejada.
- c) **mainTester**: Programa principal que executa captura de amostras de maneira interativa com o operador. Sua principal função é testar as bibliotecas de apoio, executar a calibragem dos sensores e realizar simulações independentes de missões de voo ou do controle direto da aeronave.

Para apoiar os ensaios e validar resultados, algumas aplicações externas foram desenvolvidas como Ferramentas Externas (independentes de bibliotecas ou outro programa). Sua função visa facilitar a análise dos dados obtidos e calibrar sensores, principalmente para ajustar os parâmetros do algoritmo de cálculo de distância.

- a) **BTavgDist** e **WFavgDist**: Responsáveis por ajudar na calibragem dos sensores bluetooth e WiFi. Ajudam a definir os atributos A0 (RSSI médio a 1m de distância) e C (constante ambiental) utilizados no algoritmo *Lognormal Shadow Mode* (ABDULQADER HUSSEIN et al., 2015).
- b) **printMaps**: Imprime em mapa todos os pontos obtidos numa missão de voo e o raio de distância encontrado em cada um deles. Apresenta os resultados em um arquivo html integrado com o javascript compatível com a API Googlemaps ou em tela, utilizando a biblioteca Basemaps do Python. Pode ser utilizado para validar a distância entre o ponto que representa o alvo real e o ponto estimado pelo algoritmo de trilateração, como mostrado na figura 3.7.
- c) **plotResults**: Lê o arquivo de saída JSON (*JavaScript Object Notation*), ou matrizes de resultado, e apresenta alguns gráficos predeterminados para ajudar na visualização dos dados e seus erros.

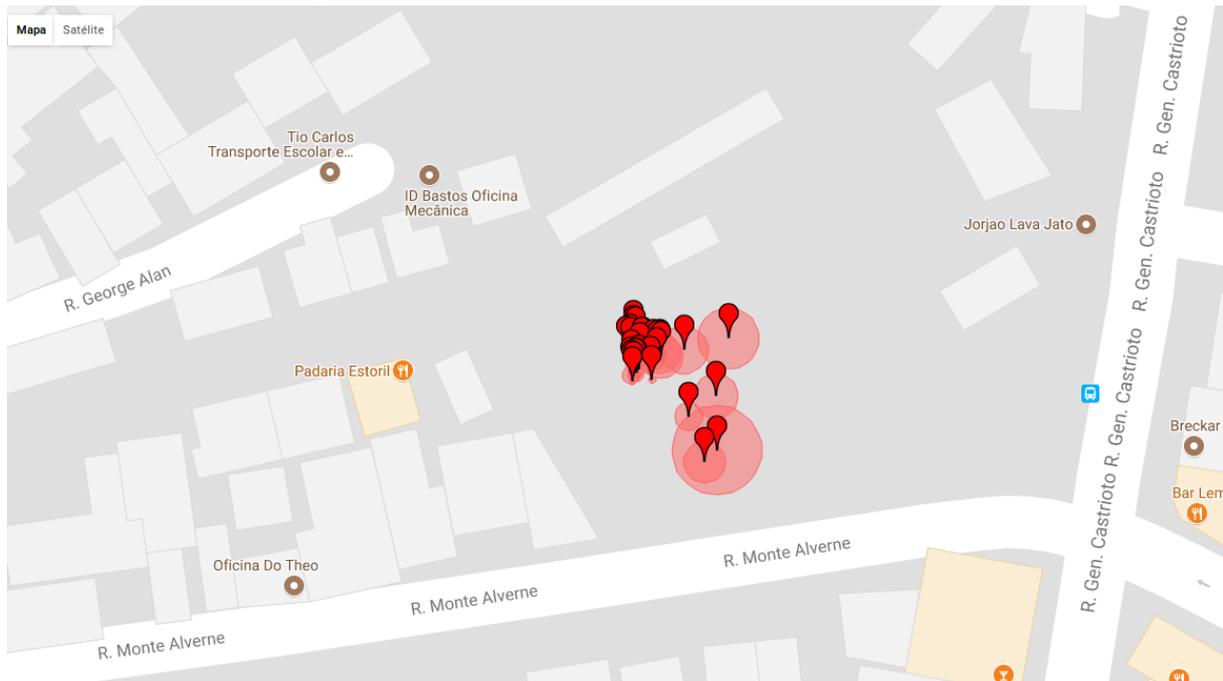


FIG. 3.7: Exemplo de saída em mapa do Google dos resultados de um voo

As Dependências Externas de *software* das aplicações desenvolvidas estão relacionadas à manipulação dos *drivers* dos sensores bluetooth e WiFi e da comunicação com o serviço Ardupilot, responsável pela operação e navegação do *hardware* de piloto automático. Abaixo são exibidas algumas dependências:

- a) **Dronekit**: *framework* de operação do Ardupilot por mensagens do padrão MAVLink para python;
- b) **WiFi**: biblioteca de controle de interfaces WiFi para Python;
- c) **Bluez**: biblioteca de controle de interfaces bluetooth para C++ e Python;
- d) **numpy**: biblioteca de funções matemáticas para Python;
- e) **basemap**: biblioteca de manipulação de dados georeferenciados para Python;
- f) **GNUplot**: biblioteca para manipular gráficos em Python, C e outras linguagens;
- g) **pyCamera**: biblioteca para operação de câmera no Raspberry Pi;
- h) **SQL Lite**: SGDBD (Sistema de gerenciamento de banco de dados) leve.

Cada biblioteca faz uso de um tipo de saída (*Output*) dependendo do parâmetro apontado pelo programa principal que a chama. Com exceção do STDOUT (*Standard*

*Output*) que é gerado por todos os componentes, quando a saída é configurada para a tela do terminal, com a finalidade de depurar a execução. O principal **Formato de Saída** para os sensores de radiofrequência na aplicação é o JSON por ser um formato de arquivo texto, de fácil leitura, com propriedades de interoperabilidade e com ferramentas eficientes para conversão e leitura da estrutura de dados nela registrada.

Na figura 3.8, a caixa Bibliotecas de Localização contém as ferramentas mais importantes da metodologia, que trazem o resultado de cada sensor, realiza a seleção dos cálculos de distância válidos, fusão e por fim trilateração dos resultados obtidos. Seu controle é determinado por parâmetros inseridos pelas aplicações principais em tempo de execução. Nesse mesmo diagrama, pode-se observar a coletânea de funções Bibliotecas do *Drone*, que representa todos os métodos utilizados para se conectar, operar e observar os atributos gerados pelo serviço Ardupilot.

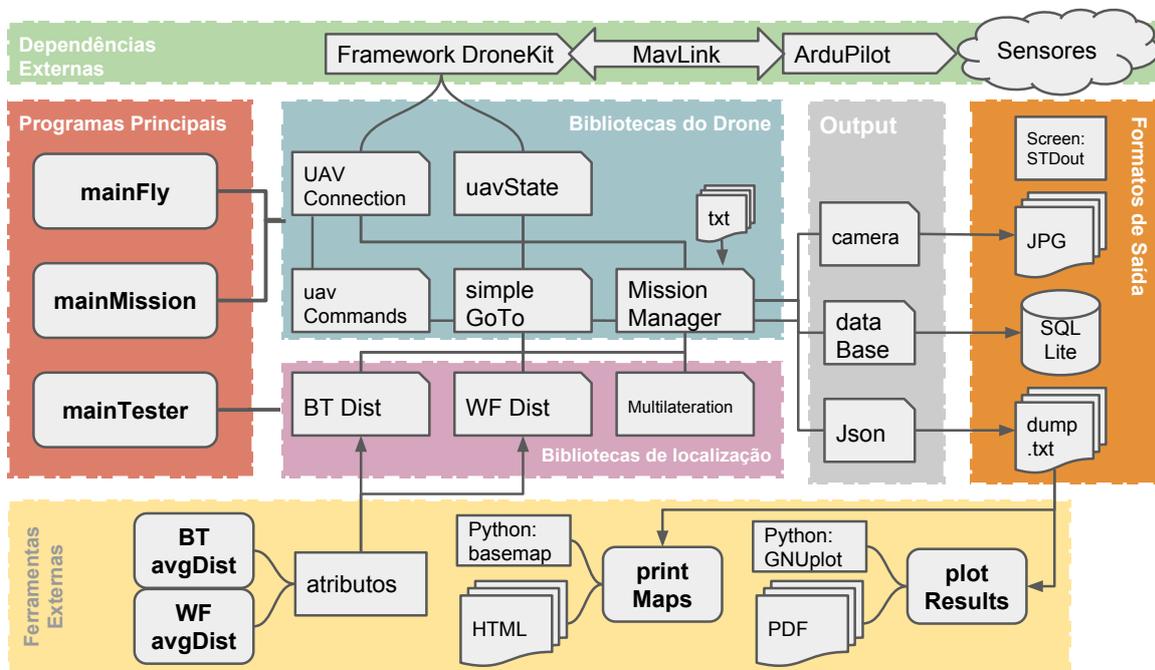


FIG. 3.8: Relação entre componentes de *software*

O controle da aeronave por *software* é determinado a partir das aplicações que fazem uso do protocolo de comunicação MAVLink. As aplicações principais desenvolvidas são capazes de chamar as bibliotecas de coleta (coletores) que por sua vez registram o resultado em um determinado formato de saída. O código principal também é responsável pela comunicação com o serviço Arducopter para obtenção de dados da aeronave, como estado dos motores e sensores de navegação, ou para operar diretamente o veículo. Essa interação é possível através do SDK (*Software Development Kit*) do Ardupilot, provido

pelo *framework* Dronekit.

O mesmo serviço Ardupilot também pode ser manipulado por agentes externos através do protocolo MAVLink. Usualmente esse recurso é utilizado por Estações de Controle em Solo. A comunicação entre um GCS e o serviço Ardupilot pode ser estabelecido por um link normal de dados WiFi (IEEE 802.11) por interface de rede ou por um rádio no padrão SiK ligado na porta UART da placa Navio2, como o 3DRobotics 433MHz, para maior resiliência de comunicação.

Por sua vez, a placa de piloto automático, Navio2 da Emlid, recebe as instruções do serviço Ardupilot no padrão MAVLink e opera os atuadores através de sinalizações no padrão PWM, enviando comandos aos ESCs que controlam os rotores da aeronave. A placa de piloto automático também pode receber dados de sensores externos pela porta I2C, como sonares e detectores LIDAR (*Light Detection and Ranging*). Se devidamente pareados, receptores de rádio controle no padrão PPM podem enviar comandos diretamente para a placa Navio2, como mostrado na figura 3.9.

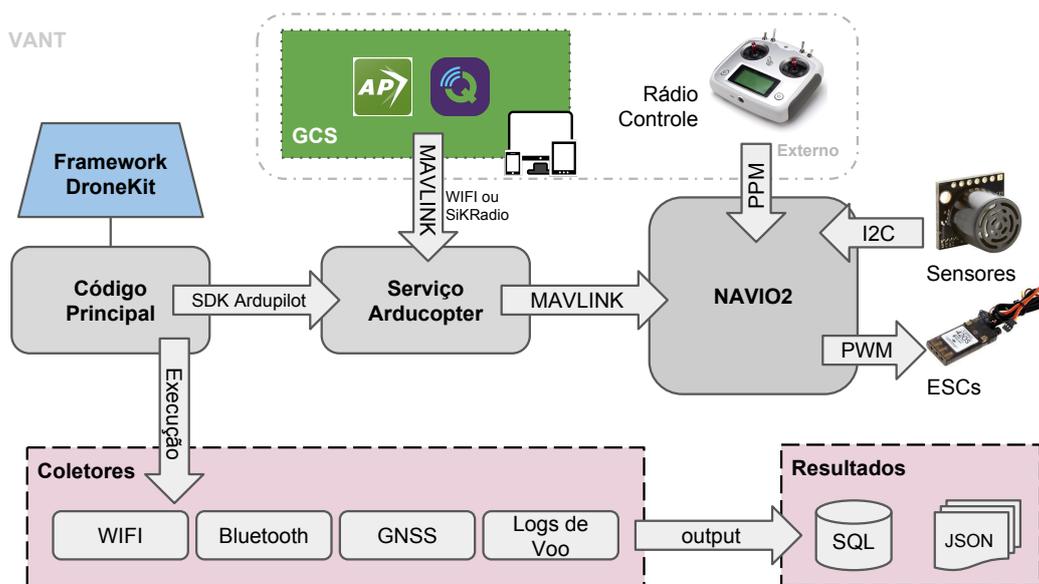


FIG. 3.9: Fluxo de controle da aeronave por *software*

Todos os códigos fontes gerados foram desenvolvidos em Python e disponibilizados em repositório público no servidor gitHub (<https://github.com/igorprata>). A solução pode ser instalada e executada em qualquer plataforma que rode Dronekit e atenda às demais dependências descritas no Anexo 8.6. Esses códigos foram validados em ensaios em campo aberto, para maior controle, utilizando apenas o sistema computacional que será embarcado (placas Navio2 e Raspberry Pi 3) em um protótipo, como o demonstrado na figura 3.10, alimentado por uma bateria de Lítio.

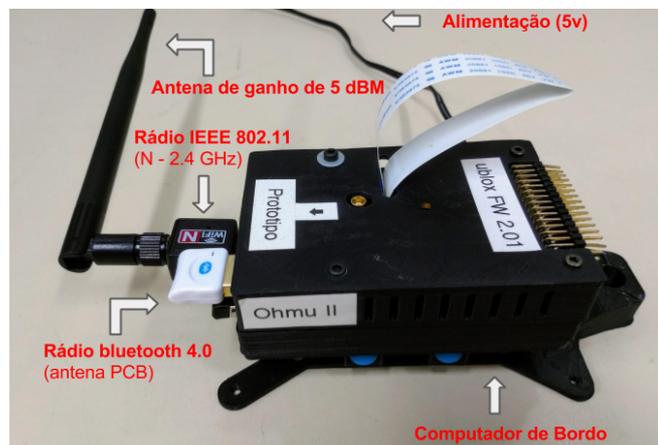


FIG. 3.10: Protótipo do sistema computacional embarcado com os sensores.

## 4 ENSAIOS

Para atestar a aplicabilidade da metodologia, foram executados diversos experimentos sob diferentes condições, com a finalidade de verificar o comportamento das variáveis necessárias ao problema de localização.

Um sistema de localização embarcado em um veículo autônomo deve estar condizente com as restrições do dispositivo em relação ao tempo de execução, consumo e processamento. A posição do VANT que funcionará como mecanismo de sensoriamento em relação ao alvo emissor de sinais de radiofrequência também deve ser considerada, para que as estimativas sejam mais precisas. Perturbações no meio e o comportamento da coleta de sinais com o deslocamento do sensor, também devem ser medidas para uma verificação apurada do erro a cada cálculo de distância.

Para evitar mais imprecisões, tanto os transmissores quanto receptores deverão estar a uma determinada altura do solo (G. BRANCO; BORDIM, 2017) para evitar interferências, como reflexão de sinais. Para aferir o cálculo de distância estimada com a distância real do transmissor, o VANT foi equipado com um ponteiro laser vermelho orientado na direção do solo. Dessa forma, medidas nesse plano, entre o ponto laser e o centro do tripé onde o alvo foi montado, puderam ser tomadas com o auxílio de uma fita métrica (Imagem 4.1).

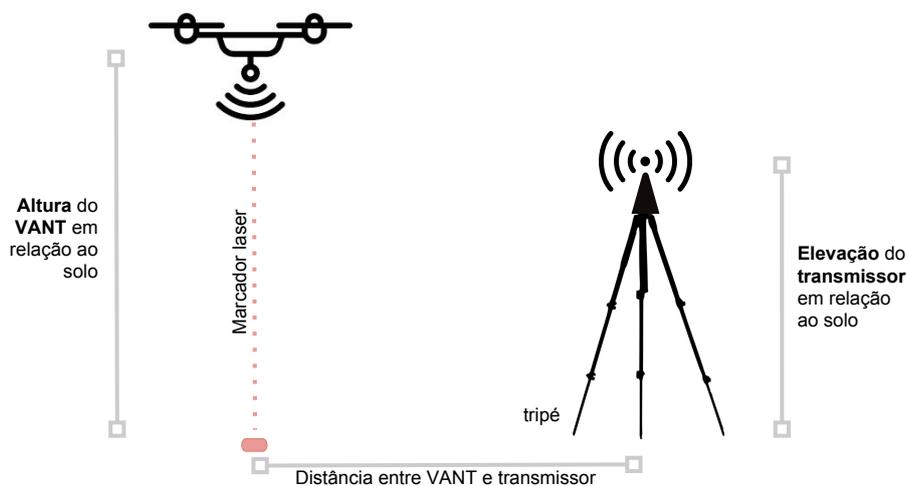


FIG. 4.1: Esquema para medições reais entre um VANT e um transmissor de RF

Com o intuito de buscar a melhor qualidade das amostras, na fase experimental foi

projetado um cenário ideal a fim de verificar a aplicabilidade da metodologia. Dessa maneira, originalmente projetou-se um ensaio de coleta de múltiplas amostras de sinais a cada ponto, de forma que o *drone* pause o seu voo para a captura de sinais e depois prossiga ao próximo ponto. Pelo *framework*, o subsistema de trilateração, por se tratar de um algoritmo de otimização, é capaz de determinar a distância de um alvo independentemente do número de pontos de verificação e amostras (a partir de 3 pontos de referência), como exemplificado na figura 4.2. Dessa forma, quanto mais pontos coletados, maior será a precisão da estimativa de localização final.

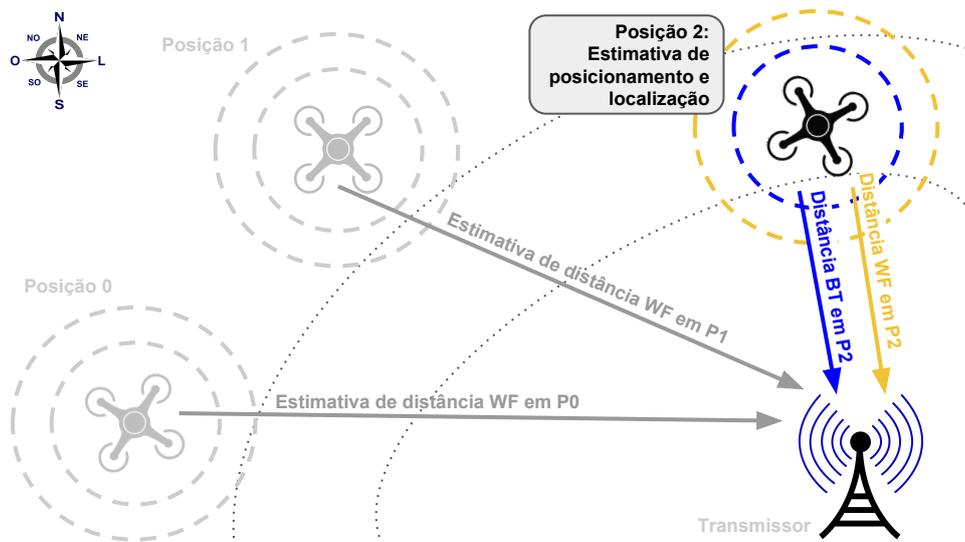


FIG. 4.2: Posições de medições de distância entre um VANT e um transmissor de RF e o momento da estimativa de localização

Devido as diferenças de especificação de cada rádio, como visto no anexo 8.4, espera-se detectar uma diferença de sinal perceptível entre os pacotes dos protocolos IEEE 802.11 e bluetooth recebidos. A análise dessa diferença e do seu comportamento em diferentes cenários, para melhor calibração dos algoritmos de estimativa de distância, também é tratada nesse capítulo.

#### 4.1 ENSAIOS INICIAIS

A primeira etapa de ensaios apresentou como resultado o desenvolvimento do *framework* de *software* que foi utilizado pela metodologia proposta, já apresentado na seção 3. Os experimentos se concentraram no teste de ferramentas de *wardriving* para coleta de amostras de sinais de radiofrequência sobre diversas condições, para validar quais atributos poderiam ser utilizados como parâmetros para localização e identificação de um alvo. Nesse

momento, também foi avaliado o desempenho das coletas, como efetuar o registro das amostras e posicionamento dos sensores em relação ao solo e outros obstáculos.

Durante os ensaios de coleta de sinais em ambientes controlados foram testadas a aplicabilidade das bibliotecas de coleta WiFiList e BTList, para captar os sinais de radiofrequência emitidos por um aparelho celular, Samsung Galaxy Win 2 (modelo SM-G360M) em um ambiente *outdoor*. Como sinais de radiofrequência são facilmente atenuados por interferências como reflexão, tanto o computador de bordo do VANT (utilizado como o protótipo apresentado na figura 3.10) e o aparelho celular, foram colocados sobre tripés a 1,4m de altura do chão. Essa decisão foi tomada para evitar que o sinal emitido pelo próprio rádio do alvo fosse refletido pelo material de revestimento do piso.

Foram feitas 3 massas de testes, uma a 1m de distância do ativo emissor de sinais e as demais à 3m e 5m (vide figura 4.3). Além da captura de sinais de bluetooth e WiFi, foram coletados os dados dos sensores de aceleração, bússola, termômetro, barômetro e giroscópio, embarcados na placa Navio2. Os dados contidos nessa primeira bateria de testes tiveram finalidade de estudar a possibilidade de publicação de uma massa de testes de localização para a comunidade científica dentro de um formato reutilizável entre diferentes projetos e melhorar a aplicabilidade geral dos arquivos de *dump* gerados.



FIG. 4.3: Ensaio de coleta apenas com o computador de bordo

## 4.2 ENSAIOS EM AMBIENTES CONTROLADOS

Por se tratar de um veículo aéreo não tripulado customizado de uso experimental, e para estar em aderência às regras da ANAC e DECEA (DECEA, 2016) os ensaios de voo exigem um espaço apropriado, com proporções suficientes para a realização da missão e onde haja garantia de segurança para as pessoas e propriedades na área. Um piloto qualificado também se faz necessário, de modo que se pudesse assumir controle do VANT autônomo e tomar ações de contingência a fim de mitigar riscos. O local de voo deve contar com boa visibilidade aos sinais de satélites e não ter obstáculos que obstruam as ondas de radiofrequência que se deseja medir, e não causar alterações súbitas nos resultados. Por esse motivo, poucos espaços dentro de uma área de conurbação atendem aos requisitos mínimos. Para oferecer as condições necessárias para esse projeto, foram realizadas diversas viagens para regiões distantes da cidade ou para campos de voo de aerodelismo. Porém, a logística envolvida neste deslocamento afeta drasticamente a rotina de ensaios. Dentro dessa restrição, nas primeiras etapas do desenvolvimento da metodologia, diversos testes em ambientes controlados foram executados. Cada experimento atendeu a um aspecto da pesquisa para que os ensaios de voo pudessem ser realizados com o máximo de confiabilidade e rendimento possível. Os testes foram divididos em simulação de voo por software, coleta de sinais em ambiente controlado e teste de voo restrito.

## 4.3 AMBIENTE DE SIMULAÇÃO DE MISSÕES DE VOO

A fim de proteger a integridade da aeronave durante o ensaio prático, foi adotado um ambiente simulado que pudesse aferir a execução do código responsável pelo controle de navegação bem como os instantes de execução de coleta de amostras.

Pela adoção do serviço Ardupilot para controle do sistema de navegação embarcado (Navio2), foi utilizado o pacote de aplicações *Software in the Loop* (SITL) oferecido pela própria comunidade mantenedora da aplicação. Essa abordagem eliminou a necessidade de um *hardware* dedicado para testes e possibilitou a validação em solo dos *softwares* embarcados, sob diversas condições.

Escrito em C++, o SITL roda nativamente em ambientes Microsoft Windows e GNU Linux (TEAM, 2016). A aplicação oferece um conjunto de ferramentas que simula todos os subsistemas embarcados no computador de bordo montado para o VANT, o que inclui uma instância completa do serviço Ardupilot.

O veículo utilizado pode ser completamente parametrizado no simulador, incluindo o número de rotores, autonomia, seu volume e distribuição de massa para um ensaio

mais fidedigno. O SITL também permite a customização de aspectos ambientais como o terreno onde o veículo irá percorrer e condições climáticas como velocidade do vento e sua direção.

Todos os comandos enviados ao VANT pela estação de controle de solo ou pelo controle remoto do piloto, também podem ser simulados no ambiente que utiliza uma camada de abstração chamada de MAVProxy (TRIDGELL, 2017). Desenvolvida em Python, o objetivo do MAVProxy é se comportar como uma instância de controle de solo minimalista que suporte toda a especificação do protocolo MAVLink. Desenvolvido originalmente pela CamberraUAV, é um software livre registrado sob a licença GNU General Public License v3.

Executando uma instância completamente funcional do Ardupilot, o SITL é capaz de suportar outros motores de simulação para melhorar a qualidade de diversos aspectos físicos, como JSBSim (BERNDT, 2017) e pode apresentar um ambiente virtual para visualização do ensaio pelo programa FlightGear (OLSON, 2017). Através do MAVProxy operando pelo protocolo UDP, outras estações de controle de solo podem ser utilizadas durante os ensaios, como o APM Planner2 (BONNEY, 2017) e o QGround Control (PROJECT, 2017) que podem interagir com o agente simulado como se fosse uma aeronave real (vide figura 4.4).

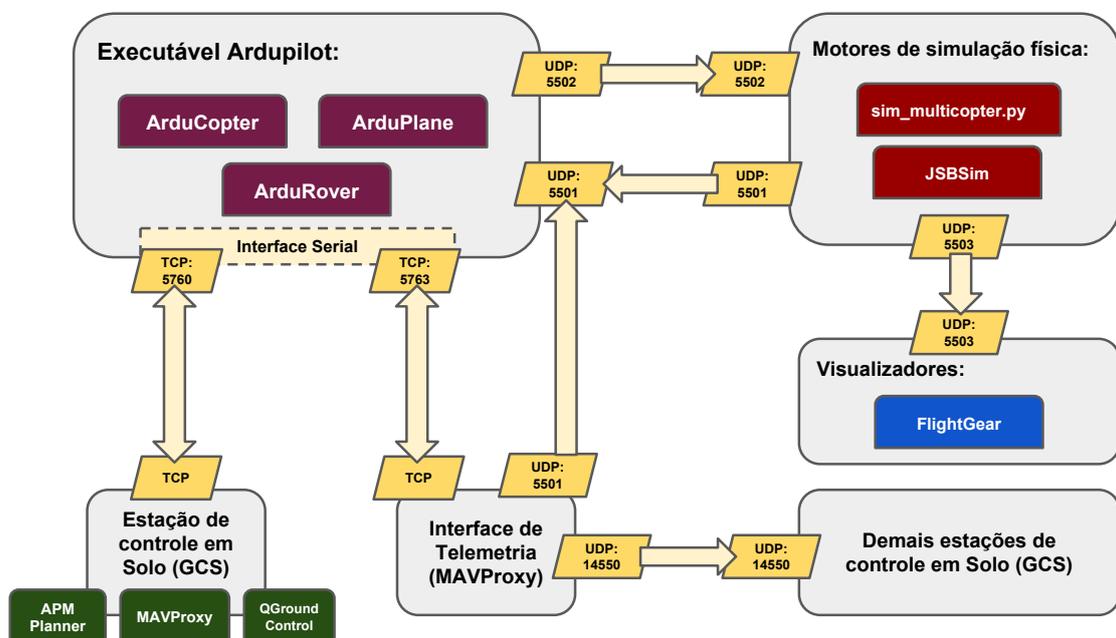


FIG. 4.4: Fluxo de controle dos componentes envolvidos na solução de *Software in the loop*

No ambiente de laboratório, foi utilizado para os ensaios um laptop HP Pavilion



aeronave por um código dedicado e a correta execução dos coletores no local esperado, e não a qualidade dos sinais de rede sem fios obtidos.



FIG. 4.6: PROTÓTIPO em execução local durante simulação de voo

#### 4.4 ENSAIOS DE COLETA DE SINAL

Para validar o código executado durante as operações em voo, além das simulações de *software in the loop*, foram realizados testes práticos de coleta de sinal em ambiente controlado. Os ensaios tiveram como objetivo medir a performance durante as coletas de sinais e a qualidade das amostras obtidas em um ambiente ideal, livre de interferências, e em cenários com ruídos, sistematicamente colocados, provocados pelo *hardware* do *drone*.

Para a construção de uma massa de dados real, durante os ensaios foram coletadas informações de RSSI de rádios bluetooth e WiFi. Aproveitando o *hardware* embarcado na placa de navegação NAVIO2, também foram coletadas informações de pressão atmosférica pelo barômetro, orientação por duas bússolas digitais, temperatura e posicionamento pelo sistema de GNSS (*Global Navigation Satellite System*).

##### 4.4.1 TESTES PELO COMPUTADOR DE BORDO

Essa etapa dos testes de coleta de sinal utilizou os mesmos componentes do computador de bordo designado para o VANT, agindo como agente receptor e um aparelho celular modelo Motorola MotoZ (XT1650), como agente emissor. No agente receptor foi utilizando

um rádio bluetooth 4.0 da Broadcom BCM20702A0 (com antena interna omnidirecional) e um rádio WiFi TP-Link Archer T2UH (embarcando um *chipset* RALINK com suporte a IEEE 802.11 AC) com uma antena externa de ganho de 2dBm, omnidirecional). Todos os rádios foram conectados pelas interfaces USB providas pelo *hardware* da placa RaspBerry Pi 3, que por sua vez foi alimentada por uma bateria externa de LiON de 8000mA/h. O agente emissor utilizou os rádios disponíveis no *hardware* embarcado no SOC Qualcomm MSM8996 Snapdragon 820: bluetooth 4.1 e WiFi no padrão IEEE 802.11 AC com suporte à dupla banda. As definições detalhadas de cada rádio empregado nas etapas dos experimentos podem ser consultadas no Anexo 8.4.

A fim de evitar interferências causadas por rebatimentos e obstrução de ondas de radiofrequência, a coleta de sinais foi validada posicionando os agentes receptor e emissor na mesma elevação em relação ao solo: 1,5m. Os ensaios foram executados em uma área aberta, para recepção de sinais do sistema GNSS, com um raio de 10m livre de paredes, como visto na figura 4.7. Foi garantido um ângulo de visão direto entre os rádios emissores e receptores, livre de obstrução em todos os pontos de coleta para evitar o efeito de *multipath*. As distâncias utilizadas na coleta das amostras foram: 1m, 3m, 5m e 10m. Para aferir a distância entre os rádios transmissores, foi utilizada uma fita métrica no plano do solo e um pêndulo para garantir a transposição da posição de cada rádio nesse plano.



FIG. 4.7: Ensaio de coleta com o VANT em ambiente controlado e sem voo

Os sinais coletados apresentaram baixa variação média uma vez que o ambiente se

mostrou próximo ao ideal, sem muitos obstáculos e interferências causadas por outras fontes emisoras na mesma frequência e canal. Observa-se pelo gráfico da figura 4.8 diferenças consistentes dependendo da fonte de alimentação utilizada e do mau dimensionamento da antena utilizada para o rádio transmissor.

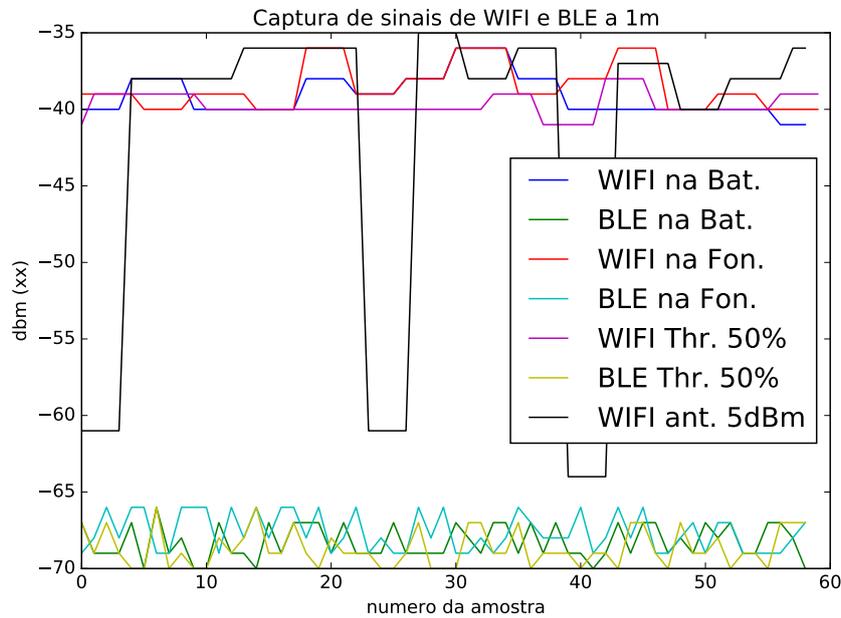


FIG. 4.8: Coletas de sinais WiFi e BLE com o VANT a 1m de distância do emissor e diferentes fontes de alimentação e antenas

#### 4.4.2 TESTES COM O VANT SEM HÉLICES

Com o objetivo de estudar as possíveis interferências causadas durante um voo pelo *hardware* de bordo de um VANT, foram desenvolvidos ensaios controlados adicionando motores, circuitos de potência de alta amperagem e a fonte de alimentação adotada com o intuito de prever possíveis alterações no algoritmo embarcado.

Os testes se basearam na aplicação prática de todo o *hardware* do VANT já montado incluindo computador de bordo, alimentação elétrica, motores e sensores, e do sistema operacional embarcado em execução. De forma a garantir a segurança do ambiente foram retiradas as hélices dos rotores, figura 4.9.

O veículo foi posicionado a 1m do plano do solo bem como o agente transmissor durante as medições em ambiente controlado. Assim como os testes de coleta de sinal, o *hardware* empregado para o agente emissor e transmissor foram os mesmos. Foram efetuadas coletas de radiofrequência a 1m do agente emissor, com o ângulo de visão livre de obstáculos (vide tabela 4.1). Cada posição recebeu 20 amostras, para o cálculo de

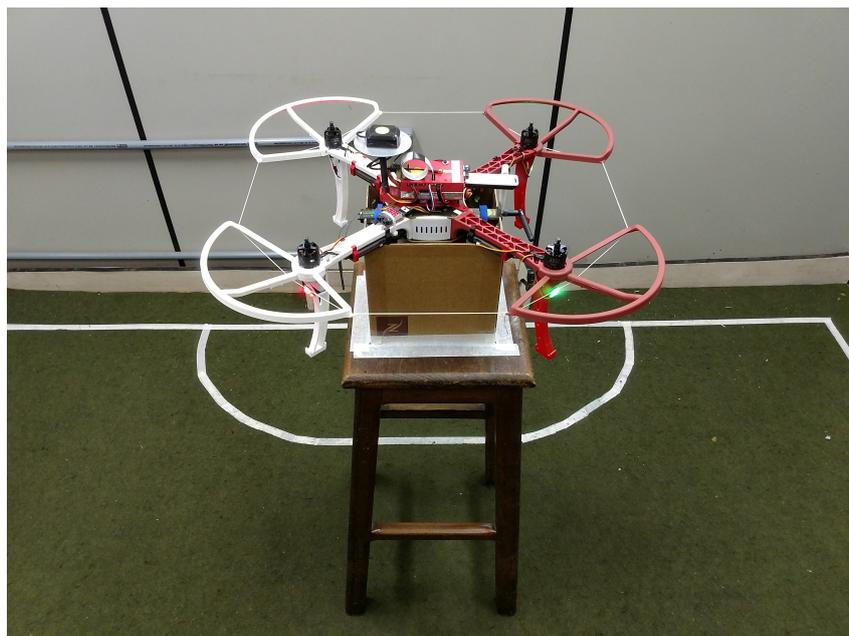


FIG. 4.9: Ensaio de coleta com o VANT em ambiente controlado com os rotores ativados

distância, com variações de potência de *throttle* nos 4 rotores de 25%, 50% e 75%, com o objetivo de medir o impacto do campo magnético dos motores, e circuito de potência, nos rádios do agente receptor. Foram também comparados resultados com diferentes fontes de alimentação sem o uso de motores. Os resultados serviram para definir parâmetros de calibração no algoritmo de *lognormal shadowing model* para o teste de voo e quais atributos interferem positivamente e negativamente nas amostras de RSSI.

TAB. 4.1: Tabela com o cálculo de distâncias efetuado a partir de amostras a 1m de distância do alvo

Número de amostras	Distância (cm)	Erro (%)	Observações
20	102,18	2%	Sem ruído dos rotores e alimentação externa
20	147,19	47%	Sem ruído dos motores e alimentação por bateria
20	94,60	5%	Motores à 25% de <i>throttle</i>
20	87,70	12%	Motores à 50% de <i>throttle</i>
20	88,05	12%	Motores à 75% de <i>throttle</i>

Durante o ensaio, o VANT foi alimentado por uma bateria Li-Po 3S de 5200mAh e 11,1 Volts e a gradação do *throttle* aplicado foi feito pelo rádio controle e os *scripts* de coleta foram disparados a cada variação. Pela figura 4.10, percebe-se um comportamento indiferente da coleta de sinais à potência de *throttle* inferida nos rotores da aeronave. Esse comportamento pode diferenciar em veículos com motores maiores e circuitos de

alimentação que superem os 20A suportado pelos ESCs do VANT utilizado no ensaio.

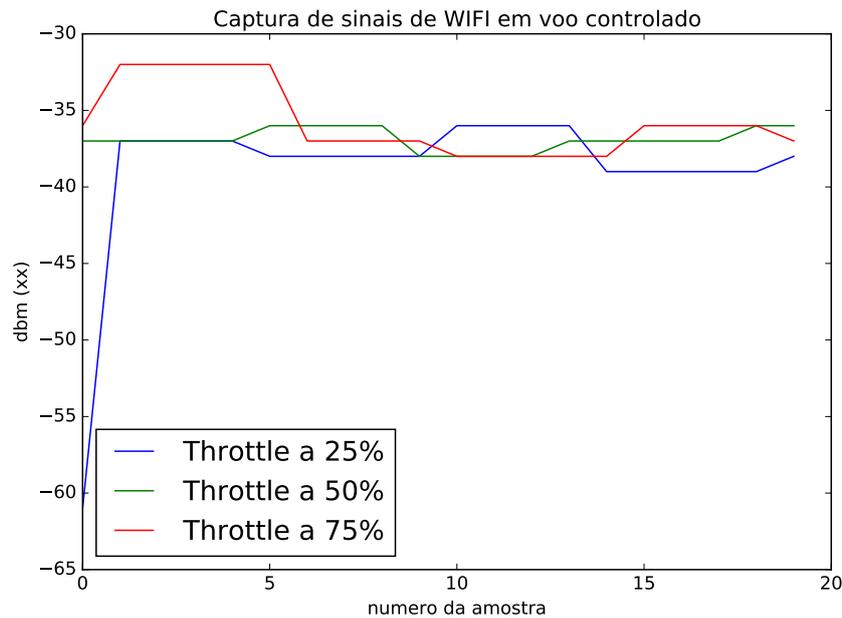


FIG. 4.10: Coletas de sinais WiFi com o VANT a 1,72m de distância do emissor e diferentes potências de *throttle*

#### 4.4.3 TESTES COM O VANT EM VOO CONTROLADO

Com a finalidade de aferir o ruído e outras perdas inerentes à execução das coletas de sinais de radiofrequência por uma aeronave em operação foi executado um teste de voo controlado em área interna. Os motores e ESCs operaram com carga suficiente para manter o *drone* estabilizado no ar. Foi determinado um ensaio prático, mas sem voo apropriado, que contou com uma estrutura elástica de suporte de modo a mitigar riscos associados, como mostrado na figura 4.11. Desta maneira, o VANT foi capaz de executar as operações de planar, elevar altitude e pousar com as hélices acopladas. Esse tipo de ensaio serviu para validar interferências oriundas de ruído eletromagnético variável (devido as mudanças de potência do motor, necessárias para manter a estabilidade) e verificar se vibrações mecânicas também seriam capazes de prejudicar a coleta de sinais.

Os testes foram realizados sob intervalo de tempo reduzido devido a restrições na alimentação elétrica. O VANT foi energizado por uma bateria Li-Po 3S de 5200mAh e 11.1 volts e a gradação do *throttle* aplicada foi feita pelo rádio controle. Os *scripts* de coleta foram executados ao se observar a estabilidade de voo a 1,5m de altura. Como observado na figura 4.12, o sinal WiFi capturado sofre um decréscimo no sensor durante o tempo de voo, provavelmente inerente à queda de tensão da bateria pela operação contínua

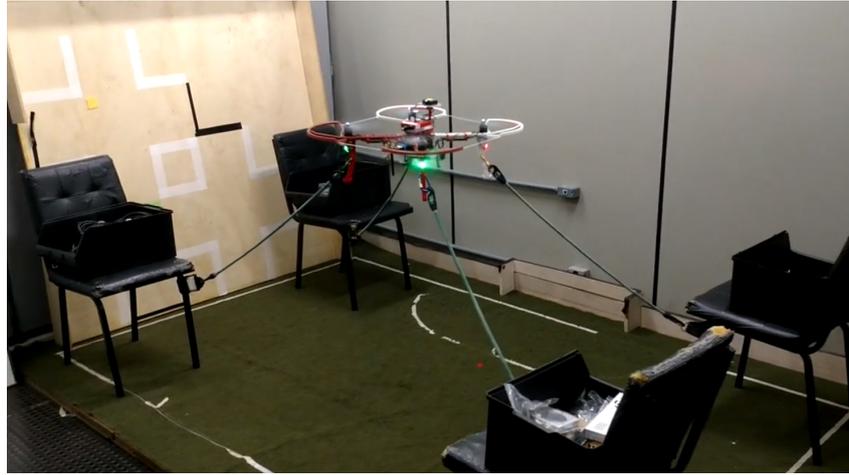


FIG. 4.11: Ensaio de coleta com o VANT em voo controlado

por alguns minutos durante um voo real.

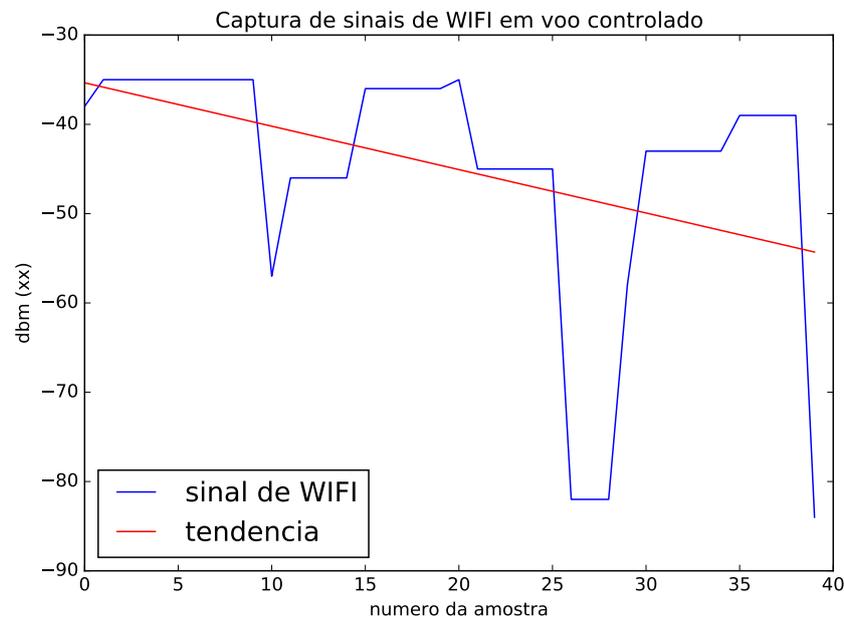


FIG. 4.12: Coletas de sinais WiFi com o VANT em voo controlado a 1,72m de distância do emissor

#### 4.5 TESTES EM MISSÕES DE VOO AUTÔNOMO

Com a homologação dos ensaios de coleta de sinais e de voo controlado, várias alterações nos *softwares* e *hardware* puderam ser feitas. O período de validação experimental pôde dessa maneira reduzir retrabalhos em campo de voo, onde as condições são mais precárias. Em especial, ao tempo de execução da aplicação de coleta de sinais que originalmente levavam 150 segundos por *waypoint*, caindo para 89 segundos por ponto após as otimizações

aplicáveis no código embarcado. Como visto na tabela 4.2, os ensaios de voo foram determinados pela quantidade de pontos de coleta desejados, o tempo de voo programado em uma rota pré-definida e o tempo total, dependente da fonte de alimentação, o que em baterias de LiPo 3S é de 12 minutos, alcançando até 15 minutos em baterias 4S na aeronave projetada.

TAB. 4.2: Tabela com o tempo de execução de coletas por número de amostras em missão de voo similar a referenciada pela tabela 7.5

Número de amostras WiFi e bluetooth	Tempo de coleta (s)	Tempo total de voo (s)	Observações
3	267	387	Aplicável em baterias 3S e 4S
4	356	476	Aplicável em baterias 3S e 4S
6	534	654	Aplicável em baterias 3S e 4S
8	712	832	Aplicável em baterias 4S ou superiores
10	890	1010	Não aplicável em baterias 3S ou 4S

Foram realizados 6 testes de voo, divididos entre as seguintes regiões:

- a) Clube de aerodelismo Asas do Tinguá (Duque de Caxias - RJ)
- b) Centro de Instrução de Operações Especiais (Niterói - RJ)
- c) RPN Cafundó (Cachoeiro de Itapemirim - ES)

Esses locais foram escolhidos devido a disponibilidade em receber testes de voo experimentais sem oferecer riscos às pessoas e às edificações no entorno. No caso do Clube Asas do Tinguá, pela disponibilidade de um piloto de aerodelismo profissional para auxiliar nos testes. A cada ensaio são seguidos os passos descritos no manual de Procedimentos Operacionais (Anexo 8.2) e a verificação dos itens básicos descritos no *check-list* de ferramentas (Anexo 8.1), de modo a evitar imprevistos antes do deslocamento para o local de voo.

Os testes de voo autônomo consistem em uma missão previamente montada em uma estação por um programa de GCS, onde são determinados os pontos que se deseja percorrer e o tempo destinado a cada coleta de amostras. Por sua vez, o arquivo com a missão de voo é carregado na aeronave com auxílio do próprio GCS ou da aplicação mainMission.py. Antes de iniciar, o VANT é posicionado com a frente na mesma direção do piloto de modo que esse não perca a orientação da aeronave no ar. Um ensaio deve ser dividido entre várias pessoas, cada uma com determinada responsabilidade no experimento:

- a) Um piloto que fica de posse do rádio controle e é capaz de tomar controle da aeronave em caso de eventos imprevistos a fim de evitar acidentes;
- b) Um operador que monitora a estação com o GCS, para acompanhar as mudanças de atributos da aeronave, o andamento da missão e os logs de mensagens;
- c) Um operador que deve acompanhar o terminal de execução do software de coleta de sinais e informar o restante da equipe em caso de falhas;
- d) Uma pessoa responsável pela segurança geral do ambiente para tomar providência em caso de trânsito de pessoas, veículo e animais, por exemplo.

Para evitar acidentes, o processo de decolagem é efetuado no modo STABILIZE pelo piloto, como observado no gráfico de estados do apêndice 7.2. Após alcançar uma altitude segura, o piloto altera o modo de operação da aeronave para AUTO, iniciando assim o voo autônomo pré-configurado. Nesse momento, a ferramenta de coleta de sinais `mainMission.py` já deve estar acionada. A aplicação aguarda a aeronave chegar a pontos específicos da missão onde a coleta de sinais pode ser feita, registrando as coordenadas de latitude e longitude e a intensidade de sinal de radiofrequência do alvo que é recebida a cada instante. A partir do terceiro ponto, o algoritmo de trilateração é executado a cada novo *waypoint*, retornando a estimativa de posicionamento do emissor a cada nova coleta efetuada.

O teste de voo utilizado para comprovar a aplicabilidade da metodologia apresentada nesse trabalho foi realizado no dia 17 de Novembro de 2017 às 10:14 da manhã no RPN Cafundó em Cachoeiro de Itapemirim no Espírito Santo. O ensaio contou com condições meteorológicas favoráveis: céu parcialmente nublado, com apenas 1% de possibilidade de chuvas e 10km de visibilidade. Previsão de ventos vindos de noroeste de no máximo 17km/h com regularidade de 6km/h e temperatura local de 26° C. A cobertura do sistema de GNSS no momento do ensaio contava com 17 satélites, destes 16 foram sintonizados, e *KP Index*<sup>1</sup> igual a 2.

A missão de voo foi bem-sucedida. Ao total foram planejados 5 pontos de coleta, cada um com 80 segundos de pausa para a captura de sinais, de um total de 14 *waypoints* como exibido na figura 4.13. Ao todo, 4 pontos dos 5 pontos inicialmente programados na missão retornaram informação aplicável na metodologia, cada ponto retornou 20 amostras de sinal WiFi e 20 amostras bluetooth.

---

<sup>1</sup>KP index: indicador que varia de 0 a 9 e denota a força dos distúrbios no campo magnético da Terra que podem influenciar na qualidade de sinal de GNSS recebido na superfície (ELLIOTT et al., 2013)



FIG. 4.13: Pontos da missão de voo de 17 de novembro de 2017 em Cachoeiro de Itape-  
mirim no ES

Todos os dados de voo podem ser consultados no apêndice 7.2. Com os pontos obtidos durante a missão de voo, a aplicação retornou os resultados apontados na figura 4.14. Os 4 pontos pretos representam os locais de coleta de sinais considerados pela metodologia, com os círculos indicando o raio das distâncias estimadas. O ponto em branco representa a posição real do alvo emissor de radiofrequência e em laranja o resultado final obtido pela trilateração com o algoritmo MSE (*Mean Square Error*). O ponto roxo exibe a posição do alvo considerando apenas o sensor WiFi em suas estimativas. O erro obtido é de menos de 3 metros o que está dentro da precisão oferecida pelo sistema de GPS com 13 satélites pareados durante o ensaio.

#### 4.6 CONCLUSÕES DA FASE EXPERIMENTAL

Em virtude de imprecisões na medida de altitude obtidas pelos sensores de barômetro e GNSS da aeronave, foi adicionado um LiDAR Garmin (modelo Lite v3) com precisão de 12cm entre 20cm e 40m de distância do alvo (GARMIN, 2016). Ao ser montado na aeronave, o sensor foi dirigido para baixo com o objetivo de manter o *drone* a uma elevação fixa em relação ao solo e ao plano do emissor de radiofrequência, melhorando a precisão dos algoritmos de trilateração em relação aos demais sensores de bordo (ver figura 4.15).

Durante a fase experimental, detectou-se necessidade de mais ensaios de voo para que a coleta de dados fosse validada em outros cenários. Dificuldades associadas à robótica aerospacial, como instabilidade do *drone* e sensibilidade a fatores externos, como

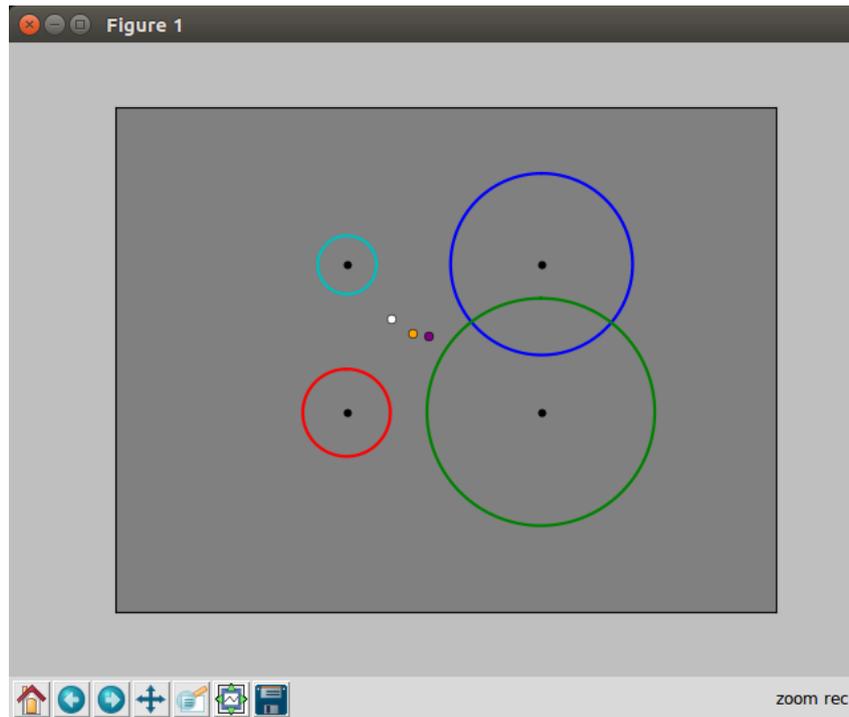


FIG. 4.14: Pontos de coleta obtidos durante o ensaio de voo de 17 de novembro de 2017 em Cachoeiro de Itapemirim no ES, com as respectivas distâncias calculadas e estimativas de localização ao centro

mudanças bruscas no clima, a dificuldade de operação humana e na própria resiliência do *software* embarcado que comanda o veículo, inviabilizaram alguns ensaios devido a pousos forçados e quedas. Todos os imprevistos foram contornados, necessitando apenas de pequenas manutenções em suas peças impressas em plástico PLA. A experiência também resultou em documentos auxiliares contendo informações sobre procedimentos operacionais, de manutenção do *hardware* e *software* (anexo 8.3, bem como uma lista de itens a serem verificadas pelo pesquisador antes de um ensaio de voo - vide anexo 8.2). Tais documentos visam mitigar os riscos associados aos experimentos práticos, principalmente em modo autônomo. Mesmo sendo executados em local seguro (longe de áreas urbanas), cada queda da aeronave representou um impedimento para o prosseguimento do experimento no mesmo dia, o que por sua vez levou a necessidade de agendar mais ensaios para compensar os anteriores.

Pela condição *sine qua non* de um ambiente seguro, afastado de conurbações e locais de tráfego aéreo em baixa altitude (como aeroportos), todos os ensaios de voo foram realizados fora da cidade do Rio de Janeiro. Esta condição de segurança e a distância a ser percorrida levou a uma mudança da logística inicial prevista para a fase de experimentos. Dessa maneira, também se buscou aderência às determinações das portarias da ANAC e

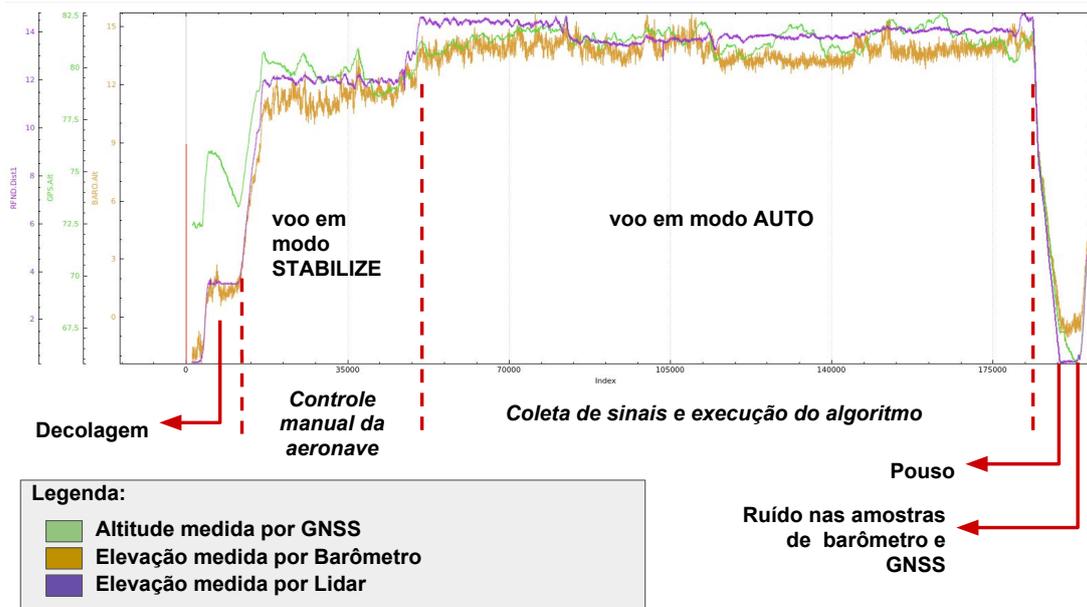


FIG. 4.15: Sensores de altitude da aeronave durante o ensaio de voo de 17/11/2017

DECEA sobre o espaço aéreo e da operação de um veículo aéreo não tripulado no Brasil.

A coleta de dados experimental obteve resultado satisfatório e pôde ser comparado com resultados obtidos em ensaios controlados como demonstrado na figura 4.16: onde os pontos de coleta utilizados na aferição de posicionamento do alvo consideraram 20 amostras cada. Foram utilizados valores de RSSI acima de -70 dBm para sinais WiFi e acima de -80 dBm para sinais bluetooth capturados. Esses valores foram processados pela biblioteca de fusão de sensores para normalização e utilizados na aferição de distância do emissor a cada ponto.

O tempo de voo máximo do VANT projetado, com a bateria designada para o ensaio e toda a carga embarcada foi de 15 minutos e 30 segundos de duração antes que o circuito de controle alertasse para a queda de tensão de cada célula de alimentação abaixo dos níveis de segurança. Dessa maneira cada ensaio de localização demandou uma bateria Li-Po 4S inteira para a realização de 4 pontos de coletas até 8 pontos de coleta de sinais, considerando manobras de deslocamento entre *waypoints* e operações de pouso e decolagem. Após cada voo, a tensão provida pela bateria foi considerada como impactante nas amostras de RSSI, necessitando reposição.

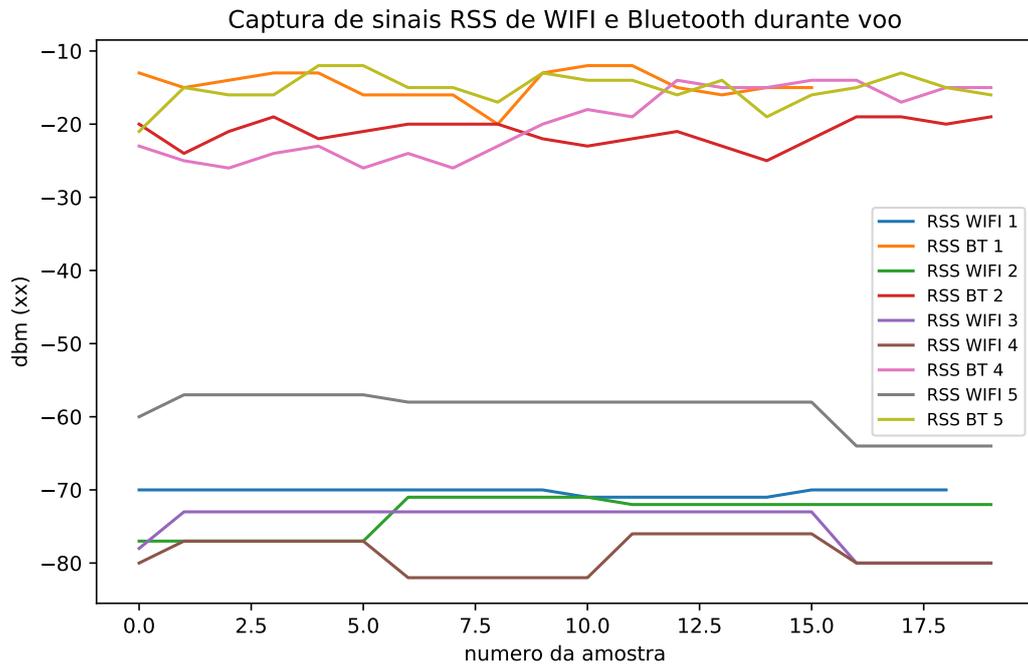


FIG. 4.16: Coleta de valores de RSSI durante o ensaio de voo de 17/11/2017



FIG. 4.17: Voo experimental sendo realizado em região agrícola no interior do Espírito Santo

## 5 CONSIDERAÇÕES FINAIS

Este capítulo é dedicado à conclusão, à descrição dos trabalhos futuros e aos agradecimentos institucionais acerca da dissertação de mestrado.

### 5.1 CONCLUSÕES

O trabalho apresentado teve por objetivo o desenvolvimento de uma metodologia implementável em recursos tecnológicos existentes, e de fácil acesso, de uma técnica de localização de alvos emissores de radiofrequência em múltiplos padrões, utilizando RSSI e trilateração. Todo o seu emprego contou com componentes de *hardware* em padrões bem difundidos, não dependendo de um fornecedor específico, e programas *open source*, possibilitando a extensão das funcionalidades da arquitetura.

Para a concepção da metodologia foi apresentada uma arquitetura escalar capaz de suportar os algoritmos através de bibliotecas de baixo acoplamento. Com a finalidade de comprovar a sua aplicabilidade, foi construída uma aeronave seguindo os padrões de projeto preestabelecidos e ensaios de voo foram realizados em campo aberto.

A metodologia se mostrou funcional em ambientes *outdoor* detectando um alvo dentro dos limites de precisão do sistema GPS utilizado para orientação de bordo do VANT (ver seção 4.5). A localização do emissor utilizou a distância calculada a partir de diversos pontos de voo e a técnica de trilateração, estimando a sua posição.

Foram detectados erros na estimativa de distância nos ensaios de voo. Essas imprecisões ocorreram principalmente em pontos de coleta afastados do emissor e durante quedas de tensão da bateria de alimentação do VANT. Esses erros estão relacionados à tolerância da técnica de força de sinal, que perde sua precisão com a distância e interferências no meio. Porém, o resultado final da estimativa foi satisfatório devido ao algoritmo de trilateração que funciona como um otimizador, calculando um ponto central às circunferências determinadas pelo raio das distâncias, mesmo quando essas circunferências não possuem interseção. Apesar de muitas estimativas de distância não serem precisas, elas ainda assim funcionaram como bons indicadores para a determinação de posição do transmissor. Quanto mais pontos de coleta forem adicionados ao sistema, e quanto melhor distribuídos ao redor do alvo forem, mais precisa será a estimativa.

A aplicação da metodologia pelas bibliotecas apresentadas nesse trabalho, funcionou

dentro de certas restrições. O padrão bluetooth 4.0, implementado na maior parte dos dispositivos, conta com radiotransmissores fracos e na maioria dos casos sem uma antena externa, gerando resultados imprecisos e com muito ruído. Sua utilização só foi considerada satisfatória dentro dos limites de 15m do raio de transmissão do alvo, acima disso os erros inseridos no algoritmo de fusão de sensores reduziram a capacidade da estimativa de distância, sendo uma opção melhor utilizar apenas o rádio IEEE 802.11.

Um sistema de localização em *Wireless Sensor Networks* precisa de um identificador para determinar o nó que se utilizará como referencial. Por praticidade, e seguindo procedimentos encontrados na literatura (KIM et al., 2012), foram utilizados apenas os endereços MAC dos dispositivos emissores como o *fingerprint* para determinação do alvo. Porém, pela facilidade de falsificação da identidade ao utilizar apenas um atributo, a metodologia comporta uma tupla de identificadores. Por questões de segurança, os valores de múltiplas informações podem ser usados para aumentar a taxa de precisão na determinação da identidade de um alvo. No caso de redes sem fio no padrão IEEE 802.11, esses valores podem ser a combinação de SSID, *MAC Address* e canal de transmissão; e em dispositivos bluetooth BLE podem ser atributos como nome e os serviços oferecidos nos pacotes de *advertisement*.

Para esse trabalho, foi optado pela detecção de redes WiFi e bluetooth por serem considerados padrões públicos de grande aceitação e acessíveis na maioria de dispositivos. Buscou-se validar a metodologia sob cenários de uso real, onde um emissor de radiofrequência poderia ser um celular a ser localizado apenas com o auxílio de seu *hardware*. Outros padrões de rádio que transmitam o valor de RSSI em seus pacotes podem ser empregados no *workflow* da metodologia com o correto desenvolvimento de bibliotecas.

Por fim, coordenadas de orientação da aeronave baseadas em informações de latitude, longitude e altitude foram retiradas da biblioteca GNSS e utilizadas como marcadores espaciais do veículo durante o voo. Essas coordenadas funcionam como um sistema local de orientação e posicionamento do VANT. Todos os cálculos de posicionamento do emissor são baseados nesses locais, onde as amostras são coletadas. Por ser um sistema de baixo acoplamento, a metodologia proposta tem o potencial de receber posições de outros sistemas de orientação. Esses podem ser utilizados para a navegação da aeronave evitando assim a dependência tecnológica e até mesmo adotando padrões com maior precisão do que os oferecidos por satélites.

## 5.2 TRABALHOS FUTUROS

Com as experiências adquiridas ao longo do desenvolvimento da metodologia proposta, foi observada a sua escalabilidade ao comportar diferentes fontes de radiofrequência e em diferentes padrões. Tomando proveito dessa característica, novos trabalhos poderão ser feitos utilizando protocolos de rádio menos usuais e em um espectro de frequência diferenciado. Essa abordagem necessitará do desenvolvimento de novas bibliotecas e calibração do algoritmo de distância para os limites detectáveis de sinais. O uso de tecnologias de comunicação exclusivas das forças armadas possibilitará a aplicação da metodologia em cenários militares especializados e com maior alcance.

A plataforma de voo não tripulada se mostrou estável para a localização de um emissor. Com o devido dimensionamento da carga paga e de sua interface com o sistema de computação embarcado, técnicas alternativas à proposta, como TDoA e AoA, podem ser experimentadas: o *chipset* de baixo custo, DWM1000 da Decawave, pode ser acoplado ao *drone* em conexão com a placa Raspberry Pi através de um adaptador USB compatível e compilando os módulos necessários ao SO. Por outro lado, um mecanismo de antenas direcionais pode ser montado sobre o maquinário responsável pela estabilização da câmera (*gimbal*) do quadricóptero, operando como um sistema rotacional controlado. Na mesma linha, para uma montagem mais simples, utilizando as informações redundantes das IMUs e bússolas digitais da Navio2, o próprio VANT poderia se comportar como uma antena direcional, rotacionando a aeronave inteira em seu eixo para a busca de diferença de sinais em diferentes ângulos.

Os resultados apresentados mostraram a detecção de um alvo em uma rota de voo preestabelecida da aeronave. Pode-se estender esse trabalho para uma operação de busca automatizada. Dessa maneira, o veículo mudaria a sua rota ao detectar a emissão de sinais até estabelecer um ponto nas proximidades do emissor. Propostas desse tipo seguem outras linhas de pesquisa, como caminho ótimo e grafos (PERAZZO et al., 2017). O VANT autônomo precisará inferir conceitos de decisão, como critério de parada e escolha de direção, que podem ser auxiliados com as estimativas geradas a cada ponto pela metodologia proposta.

Os resultados experimentais apresentados utilizaram um emissor de radiofrequência fixo de forma a não injetar outras variáveis ao problema de localização. Novos trabalhos para detectar a tolerância da metodologia para alvos errantes podem ser propostos. Desse modo, seriam estabelecidos indicadores relevantes para inferir sobre o deslocamento de um transmissor e em quais condições a estimativa de posição ainda será considerada precisa.

### 5.3 AGRADECIMENTOS INSTITUCIONAIS

Faz-se agradecimentos às seguintes instituições que viabilizaram a realização deste trabalho:

- CAPES (Comissão de Aperfeiçoamento do Pessoal do Nível Superior), por conceder uma bolsa de MESTRADO por meio do Edital PROAP / Demanda Social;
- IME (Instituto Militar de Engenharia), pela infraestrutura e fomento de produção científica dentro de seu campus.

## 6 REFERÊNCIAS BIBLIOGRÁFICAS

- ABDULQADER HUSSEIN, A.; RAHMAN, T. A. ; LEOW, C. Y. Performance evaluation of localization accuracy for a log-normal shadow fading wireless sensor network under physical barrier attacks. **Sensors**, v. 15, n. 12, p. 30545–30570, 2015. Disponível em: <<http://www.mdpi.com/1424-8220/15/12/29817>>. Acesso em: 2016-11-28.
- ABRAMOV, E.; KOBILEV, M. ; MAKAREVICH, O. Using quadrocopter as a pentest tool. In: PROCEEDINGS OF THE 6TH INTERNATIONAL CONFERENCE ON SECURITY OF INFORMATION AND NETWORKS, 6th., SIN '13, 1., 2013. **Anais...** New York, NY, USA: ACM, 2013, p. 404–407. Disponível em: <<http://doi.acm.org/10.1145/2523514.2527019>>. Acesso em: 2016-11-28.
- AIRCRACK-NG. AirCrack-NG. Disponível em: <<https://www.aircrack-ng.org/>>. Acesso em: 2016-11-28.
- ANANTHANARAYANAN, G.; STOICA, I. Blue-fi: Enhancing wi-fi performance using bluetooth signals. In: PROCEEDINGS OF THE 7TH INTERNATIONAL CONFERENCE ON MOBILE SYSTEMS, APPLICATIONS, AND SERVICES, 7th., MOBISYS '09, 1., 2009. **Anais...** New York, NY, USA: ACM, 2009, p. 249–262. Disponível em: <<http://doi.acm.org/10.1145/1555816.1555842>>. Acesso em: 2016-11-28.
- JON BERNDT. SBSim an open source flight dynamics model (FDM). Disponível em: <<http://jsbsim.sourceforge.net/>>. Acesso em: 24 jan. 2017.
- BILL BONNEY. APM Planner 2.0 is an open-source ground station application for MAVlink based autopilots. Disponível em: <<http://ardupilot.org/planner2/>>. Acesso em: 24 jan. 2017.
- GENE BRANSFIELD. Weaponizing Your Pets: The War Kitten and the Denial of Service Dog (Defcon 22). Disponível em: <<https://www.defcon.org/html/defcon-22/dc-22-speakers.html#Bransfield>>. Acesso em: 2016-11-28.
- BYCHKOVSKY, V.; HULL, B.; MIU, A.; BALAKRISHNAN, H. ; MADDEN, S. A measurement study of vehicular internet access using in situ wi-fi networks. In: PROCEEDINGS OF THE 12TH ANNUAL INTERNATIONAL CONFERENCE ON MOBILE COMPUTING AND NETWORKING, 12th., MOBICOM '06, 1., 2006. **Anais...** New York, NY, USA: ACM, 2006, p. 50–61. Disponível em: <<http://doi.acm.org/10.1145/1161089.1161097>>. Acesso em: 2016-11-28.
- CHEN, Y.; A. FRANCISCO, J.; TRAPPE, W. ; MARTIN, R. P. A practical approach to landmark deployment for indoor localization. In: 2006 3RD ANNUAL IEEE COMMUNICATIONS SOCIETY ON SENSOR AND AD HOC COMMUNICATIONS AND NETWORKS, 1st., 2006. **Anais...** [S.l.: s.n.], 2006, p. 365–373.

- CHENG, C.-C.; HSIU, P.-C. Extend your journey: Considering signal strength and fluctuation in location-based applications. **IEEE/ACM Trans. Netw.**, v. 23, n. 2, p. 451–464, 2015. Disponível em: <<http://dx.doi.org/10.1109/TNET.2014.2301171>>. Acesso em: 2016-11-28.
- CHINTALAPUDI, K.; PADMANABHA IYER, A. ; PADMANABHAN, V. N. Indoor localization without the pain. In: PROCEEDINGS OF THE SIXTEENTH ANNUAL INTERNATIONAL CONFERENCE ON MOBILE COMPUTING AND NETWORKING, 10th., MOBICOM '10, 1., 2010. **Anais...** New York, NY, USA: ACM, 2010, p. 173–184. Disponível em: <<http://doi.acm.org/10.1145/1859995.1860016>>. Acesso em: 2016-11-28.
- CRIVELLA, A. **Fusão de Sensores para um VANT via Suavização Incremental Baseada em Grafos-fatores**. 2015. 133 f. Dissertação (Sistemas e Computação) – Instituto Militar de Engenharia, Rio de Janeiro, Brasil, 2015. Disponível em: <<http://www.comp.ime.eb.br/pos/arquivos/publicacoes/dissertacoes/2015/2015-Anderson.pdf>>. Acesso em: 2016-11-28.
- DA DEFESA, B. M. Estratégia nacional de defesa. In: DA DEFESA NACIONAL, B. M. (Org.). **Estratégia nacional de defesa**. Brasília, Brasil: Ministério da Defesa, 2008. p. 64.
- AGÊNCIA NACIONAL DE AVIAÇÃO CIVIL. Drones são usados por bombeiros de São Paulo para combater incêndios. Disponível em: <<https://www.anac.gov.br/assuntos/paginas-tematicas/drones>>. Acesso em: 2016-11-28.
- AGÊNCIA NACIONAL DE AVIAÇÃO CIVIL. Drones são usados por bombeiros de São Paulo para combater incêndios. Disponível em: <<http://www2.anac.gov.br/rpas/>>. Acesso em: 2016-11-28.
- AGÊNCIA NACIONAL DE TELECOMUNICAÇÕES. Resolução nº 671. Disponível em: <[www.anatel.gov.br/legislacao/resolucoes/2016/911-resolucao-671](http://www.anatel.gov.br/legislacao/resolucoes/2016/911-resolucao-671)>. Acesso em: 2016-11-28.
- DECEA, DEPARTAMENTO DE CONTROLE DO ESPAÇO AÉREO. ICA 100-40 Sistemas de Aeronaves Remotamente Pilotadas e o Acesso ao Espaço Aéreo Brasileiro. Disponível em: <<http://www.decea.gov.br/drone/>>. Acesso em: 2017-11-28.
- DUNKLEY, O.; ENGEL, J.; STURM, J. ; CREMERS, D. Visual-inertial navigation for a camera-equipped 25g nano-quadrotor. In: IROS2014 AERIAL OPEN SOURCE ROBOTICS WORKSHOP, 1st., 2014. **Anais...** [S.l.: s.n.], 2014, p. 427–440. Disponível em: <<https://vision.in.tum.de/media/spezial/bib/dunkley14iros.pdf> > .Acessoem : 2016 – 11 – 28.
- EKHATOR, S. **Evaluating Kismet and NetStumbler as Network Security Tools & Solutions**. 2010. 89 f. Dissertação (School of Computing) – Blekinge Institute of Technology, Uppsala, Sweden, 2010. Disponível em: <<http://urn.kb.se/resolve?urn=urn:nbn:se:bth-5668>>. Acesso em: 2016-11-28.

- ELLIOTT, H. A.; JAHN, J. M. ; MCCOMAS, D. J. The kp index and solar wind speed relationship: Insights for improving space weather forecasts. **Space Weather**, v. 11, n. 6, p. 339–349, 2013.
- CORPORATION EMLID. Navio2 - autopilot HAT for Raspberry Pi. Disponível em: <<https://docs.emlid.com/navio2>>. Acesso em: 2017-07-24.
- FARAGHER, R.; HARLE, R. Location fingerprinting with bluetooth low energy beacons. **IEEE Journal on Selected Areas in Communications**, v. 33, n. 11, p. 2418–2428, 2015.
- RASPBERRY PI FOUNDATION. Raspberry Pi 3 Model B especifications. Disponível em: <<https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>>. Acesso em: 2016-11-28.
- RASPBERRY PI FOUNDATION. Raspberry Pi Foundation. Disponível em: <<https://www.raspberrypi.org/>>. Acesso em: 2016-11-28.
- G. BRANCO, S. V. P.; BORDIM, J. L. Physical web: Avaliação de técnicas de estimativa de localização de dispositivos ble em ambientes internos. In: V WORKSHOP OF COMMUNICATION IN CRITICAL EMBEDDED SYSTEMS, 5., WORKSHOP OF COMMUNICATION IN CRITICAL EMBEDDED SYSTEMS, 5., 2017, Belém. **Proceedings...** CEP 91.509-900 - Porto Alegre - RS: Sociedade Brasileira de Computação (SBC), 2017, p. 67–76.
- GLOBO G1. Desastre Ambiental em Mariana. Disponível em: <<http://g1.globo.com/minas-gerais/desastre-ambiental-em-mariana/>>. Acesso em: 2016-11-28.
- GLOBO G1. Drones são usados por bombeiros de São Paulo para combater incêndios. Disponível em: <<http://g1.globo.com/tecnologia/campus-party/2016/noticia/2016/01/drones-sao-usados-por-bombeiros-de-sao-paulo-para-combater-incendios.html>>. Acesso em: 2016-11-28.
- GLOBO G1. Número de pessoas perdidas em trilhas aumenta 20 no litoral de SP. Disponível em: <<http://g1.globo.com/sp/santos-regiao/verao/2015/noticia/2015/01/numero-de-pessoas-perdidas-em-trilhas-aumenta-20-no-litoral-de-sp.html>>. Acesso em: 2016-11-28.
- CORPORATION GARMIN. Lidar Lite v3 Operation Manual and Technical Specifications. Disponível em: <<https://static.garmin.com/pumac/LIDAR-Lite-v3-Operation-Manual-and-Technical-Specifications.pdf>>. Acesso em: 2016-11-28.
- GERCINA GONÇALVES DA SILVA, KLEBER PADOVANI DE SOUZA, A. B. G. H. P.; GONZAGA, J. F. Veículos aéreos não tripulados com visão computacional na agricultura: Aplicações, desafios e perspectivas. In: ANAIS DO ENCONTRO CIENTÍFICO DE ADMINISTRAÇÃO, ECONOMIA E CONTABILIDADE, 7., 2015. **Anais...** [S.l.: s.n.], 2015, p. 1–12.
- GÜNAY, F. B.; ÇAVDAR, T. Mobile fleet localization model via rssi, toa and tdoa in wireless sensor networks. In: 2014 22ND SIGNAL PROCESSING AND COMMUNICATIONS APPLICATIONS CONFERENCE (SIU), 1st., 2014. **Anais...** [S.l.: s.n.], 2014, p. 1431–1434.

- HAEBERLEN, A.; FLANNERY, E.; LADD, A. M.; RUDYS, A.; WALLACH, D. S. ; KAVRAKI, L. E. Practical robust localization over large-scale 802.11 wireless networks. In: PROCEEDINGS OF THE 10TH ANNUAL INTERNATIONAL CONFERENCE ON MOBILE COMPUTING AND NETWORKING, 4th., MOBICOM '04, 1., 2004. **Anais...** New York, NY, USA: ACM, 2004, p. 70–84. Disponível em: <<http://doi.acm.org/10.1145/1023720.1023728>>. Acesso em: 2016-11-28.
- HAWKINSON, W.; SAMANANT, P.; MCCROSKEY, R.; INGVALSON, R.; KULKARNI, A.; HAAS, L. ; ENGLISH, B. Glanser: Geospatial location, accountability, and navigation system for emergency responders - system concept and performance assessment. In: PROCEEDINGS OF THE 2012 IEEE/ION POSITION, LOCATION AND NAVIGATION SYMPOSIUM, 1st., 2012. **Anais...** [S.l.: s.n.], 2012, p. 98–105.
- HERFURT, M.; MULLINER, C. Remote device identification based on bluetooth fingerprinting techniques. **Trifinite White paper**, v. 0.3, 2004. Disponível em: <<http://trifinite.org/Downloads/Blueprinting.pdf>>. Acesso em: 2016-11-28.
- JENKS, C. Real-time rogue wireless access point detection with the raspberry pi. **Linux J.**, v. 2014, n. 248, 2014. Disponível em: <<http://dl.acm.org/citation.cfm?id=2714050.2714052>>. Acesso em: 2016-11-28.
- KIM, M.; FIELDING, J. J. ; KOTZ, D. Risks of using ap locations discovered through war driving. In: FISHKIN, K. P.; SCHIELE, B.; NIXON, P. ; QUIGLEY, A. (Org.). **Pervasive Computing: 4th International Conference, PERSVASIVE 2006, Dublin, Ireland, May 7-10, 2006. Proceedings**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. p. 67–82. ISBN 978-3-540-33895-6.
- KIM, Y.; CHON, Y. ; CHA, H. Smartphone-based collaborative and autonomous radio fingerprinting. **IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)**, v. 42, n. 1, p. 112–122, 2012.
- KOTARU, M.; JOSHI, K.; BHARADIA, D. ; KATTI, S. Spotfi: Decimeter level localization using wifi. **SIGCOMM Comput. Commun. Rev.**, v. 45, n. 4, p. 269–282, 2015. Disponível em: <<http://doi.acm.org/10.1145/2829988.2787487>>. Acesso em: 2016-11-28.
- KREVL, A.; CIGLARIC, M. A framework for developing distributed location based applications. In: PROCEEDINGS 20TH IEEE INTERNATIONAL PARALLEL DISTRIBUTED PROCESSING SYMPOSIUM, 20th., 2006. **Anais...** [S.l.: s.n.], 2006, p. 6 pp.–. Acesso em: 2016-11-28.
- LIN, K.; KANSAL, A.; LYMBEROPOULOS, D. ; ZHAO, F. Energy-accuracy trade-off for continuous mobile device location. In: PROCEEDINGS OF THE 8TH INTERNATIONAL CONFERENCE ON MOBILE SYSTEMS, APPLICATIONS, AND SERVICES, 10th., MOBISYS '10, 1., 2010. **Anais...** New York, NY, USA: ACM, 2010, p. 285–298. Disponível em: <<http://doi.acm.org/10.1145/1814433.1814462>>. Acesso em: 2016-11-28.
- MAO, G.; FIDAN, B. ; ANDERSON, B. D. O. Wireless sensor network localization techniques. **Comput. Netw.**, v. 51, n. 10, p. 2529–2553, 2007. Disponível em: <<http://dx.doi.org/10.1016/j.comnet.2006.11.018>>. Acesso em: 2016-11-28.

- MICHAEL MUELLER. Bluetooth Scanner - btCrawler. Disponível em: <<https://play.google.com/store/apps/details?id=com.silentservices.btCrawler>>. Acesso em: 2016-11-28.
- MUNSHI, A. Q.; MISHU, M. ; SAYEED, K. A. A low cost cospas-sarsat alternative for epirb transponder for local fishing boats in bangladesh. In: 2014 10TH INTERNATIONAL CONFERENCE ON COMMUNICATIONS (COMM), 10th., 2014. **Anais...** [S.l.: s.n.], 2014, p. 1–6.
- NEISHABOORI, A.; HARRAS, K. Energy saving strategies in wifi indoor localization. In: PROCEEDINGS OF THE 16TH ACM INTERNATIONAL CONFERENCE ON MODELING, ANALYSIS; SIMULATION OF WIRELESS AND MOBILE SYSTEMS, 13th., MSWIM '13, 1., 2013. **Anais...** New York, NY, USA: ACM, 2013, p. 399–404. Disponível em: <<http://doi.acm.org/10.1145/2507924.2507997>>. Acesso em: 2016-11-28.
- NICHOLSON, A. J.; NOBLE, B. D. Breadcrumbs: Forecasting mobile connectivity. In: PROCEEDINGS OF THE 14TH ACM INTERNATIONAL CONFERENCE ON MOBILE COMPUTING AND NETWORKING, 8th., MOBICOM '08, 1., 2008. **Anais...** New York, NY, USA: ACM, 2008, p. 46–57. Disponível em: <<http://doi.acm.org/10.1145/1409944.1409952>>. Acesso em: 2016-11-28.
- NMAP.ORG. Nmap: the Network Mapper - Free Security Scanner. Disponível em: <<https://nmap.org/>>. Acesso em: 2016-11-28.
- NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION. Search and Rescue Satellite Aided Tracking. Disponível em: <<http://www.sarsat.noaa.gov/emerbcons.html>>. Acesso em: 2016-11-28.
- CURTIS L. OLSON. FlightGear Flight Simulator. Disponível em: <<http://www.flightgear.org>>. Acesso em: 24 jan. 2017.
- PERAZZO, P.; SORBELLI, F. B.; CONTI, M.; DINI, G. ; PINOTTI, C. M. Drone path planning for secure positioning and secure position verification. **IEEE Transactions on Mobile Computing**, v. 16, n. 9, p. 2478–2493, 2017.
- PERING, T.; AGARWAL, Y.; GUPTA, R. ; WANT, R. Coolspots: Reducing the power consumption of wireless mobile devices with multiple radio interfaces. In: PROCEEDINGS OF THE 4TH INTERNATIONAL CONFERENCE ON MOBILE SYSTEMS, APPLICATIONS AND SERVICES, 6th., MOBISYS '06, 1., 2006. **Anais...** New York, NY, USA: ACM, 2006, p. 220–232. Disponível em: <<http://doi.acm.org/10.1145/1134680.1134704>>. Acesso em: 2016-11-28.
- PRIETO, J.; MAZUELAS, S.; BAHILLO, A.; FERNANDEZ, P.; LORENZO, R. M. ; ABRIL, E. J. Adaptive data fusion for wireless localization in harsh environments. **IEEE Transactions on Signal Processing**, v. 60, n. 4, p. 1585–1596, 2012.
- PRIYANTHA, N. B. **The Cricket Indoor Location System**. 2005. Tese (Department of Electrical Engineering and Computer Science) – Massachusetts Institute of Technology, Cambridge, MA, USA, 2005. Acesso em: AAI0808861.

- DRONECODE PROJECT. Cross-platform ground control station for drones (Android, iOS, Mac OS, Linux, Windows). Disponível em: <<https://github.com/mavlink/qgroundcontrol>>. Acesso em: 24 jan. 2017.
- RAHMATI, A.; ZHONG, L. Context-for-wireless: Context-sensitive energy-efficient wireless data transfer. In: PROCEEDINGS OF THE 5TH INTERNATIONAL CONFERENCE ON MOBILE SYSTEMS, APPLICATIONS AND SERVICES, 7th., MOBISYS '07, 1., 2007. **Anais...** New York, NY, USA: ACM, 2007, p. 165–178. Disponível em: <<http://doi.acm.org/10.1145/1247660.1247681>>. Acesso em: 2016-11-28.
- RAI, A.; CHINTALAPUDI, K. K.; PADMANABHAN, V. N. ; SEN, R. Zee: Zero-effort crowdsourcing for indoor localization. In: PROCEEDINGS OF THE 18TH ANNUAL INTERNATIONAL CONFERENCE ON MOBILE COMPUTING AND NETWORKING, 18th., MOBICOM '12, 1., 2012. **Anais...** New York, NY, USA: ACM, 2012, p. 293–304. Disponível em: <<http://doi.acm.org/10.1145/2348543.2348580>>. Acesso em: 2016-11-28.
- RICARDO M. BERNARDO, L. C. B. D. S.; ROSA, P. F. F. Concepção de uma plataforma de vant de baixo custo do tipo quadricóptero para uso em pesquisas. In: V WORKSHOP OF COMMUNICATION IN CRITICAL EMBEDDED SYSTEMS, 5., WORKSHOP OF COMMUNICATION IN CRITICAL EMBEDDED SYSTEMS, 5., 2017, Belém. **Proceedings...** CEP 91.509-900 - Porto Alegre - RS: Sociedade Brasileira de Computação (SBC), 2017, p. 56–65.
- RISTANOVIC, N.; TRAN, D.-K. ; LE BOUDEC, J.-Y. Tracking of mobile devices through bluetooth contacts. In: PROCEEDINGS OF THE ACM CO-NEXT STUDENT WORKSHOP, 10th., CONEXT '10 STUDENT WORKSHOP, 1., 2010. **Anais...** New York, NY, USA: ACM, 2010, p. 4:1–4:2. Disponível em: <<http://doi.acm.org/10.1145/1921206.1921211>>. Acesso em: 2016-11-28.
- 3D ROBOTICS. Arming Check importance. Disponível em: <[http://python.dronekit.io/1.5.0/guide/sitl\\_setup.html#disable-arming-checks](http://python.dronekit.io/1.5.0/guide/sitl_setup.html#disable-arming-checks)> .Acessoem : 2017 – 07 – 24.
- 3D ROBOTICS. Arquitetura do Dronekit para Python. Disponível em: <<http://python.dronekit.io/>>. Acesso em: 2017-07-24.
- 3D ROBOTICS. Compatible companion computer devices. Disponível em: <<http://python.dronekit.io/develop/companion-computers.html>>. Acesso em: 2017-11-28.
- 3D ROBOTICS. Instalação do DroneKit. Disponível em: <[http://python.dronekit.io/develop/sitl\\_setup.html](http://python.dronekit.io/develop/sitl_setup.html)> .Acessoem : 2017 – 07 – 24.
- S., J.; MILLER, G. M. **Modern Electronic Communication (9th Ed.)**. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2008. ISBN 9780132251136.
- SAPIEZYNSKI, P.; GATEJ, R.; MISLOVE, A. ; LEHMANN, S. Opportunities and challenges in crowdsourced wardriving. In: PROCEEDINGS OF THE 2015 ACM CONFERENCE ON INTERNET MEASUREMENT CONFERENCE, 15th., IMC '15, 1., 2015. **Anais...** New York, NY, USA: ACM, 2015, p. 267–273. Disponível em: <<http://doi.acm.org/10.1145/2815675.2815711>>. Acesso em: 2016-11-28.

- BLUETOOTH SIG. especificação do padrão Bluetooth. Disponível em: <<https://www.bluetooth.com/>>. Acesso em: 2016-11-28.
- SIQUEIRA, A. **Algoritmo de Localização de Rede**. 2017. 95 f. Dissertação (Sistemas e Computação) – Instituto Militar de Engenharia, Rio de Janeiro, Brasil, 2017. Disponível em: <<http://www.comp.ime.eb.br/pos/conteudo/publicacoes/lista-dissertacoes.html>>. Acesso em: 2017-12-28.
- SOURCEFORGE. bluediving - Next generation Bluetooth security tool. Disponível em: <<http://bluediving.sourceforge.net/>>. Acesso em: 2016-11-28.
- ARDUPILOT DEV TEAM. Tutorial de configuração do SITL no Linux. Disponível em: <<http://ardupilot.org/dev/docs/setting-up-sitl-on-linux.html>>. Acesso em: 2017-07-24.
- THOMAS KING, STEPHAN KOPF, W. E. A location system based on sensor fusion: Research areas and software architecture. **Proc. of 2nd GI/ITG KuVS Fachgespräch"Ortsbezogene Anwendungen und Dienste"**, Stuttgart, Germany, v. 1, p. 28–32, 2006. Disponível em: <<https://ub-madoc.bib.uni-mannheim.de/38963/1/King2005a.pdf>> .*Acessoem* : 2016 – 11 – 28.
- ANDREW TRIDGELL. MAVProxy - a UAV ground station software package for MAVLink based systems. Disponível em: <<https://github.com/ArduPilot/MAVProxy>>. Acesso em: 2017-07-24.
- TSUI, A. W. T.; LIN, W. C.; CHEN, W. J.; HUANG, P. ; CHU, H. H. Accuracy performance analysis between war driving and war walking in metropolitan wi-fi localization. **IEEE Transactions on Mobile Computing**, v. 9, n. 11, p. 1551–1562, 2010. Disponível em: <<http://ieeexplore.ieee.org/document/5499474/>>. Acesso em: 2016-11-28.
- VAUGHAN-NICHOLS, S. J. Will mobile computing's future be location, location, location?. **Computer**, v. 42, n. 2, p. 14–17, 2009. Disponível em: <<http://dx.doi.org/10.1109/MC.2009.65>>. Acesso em: 2016-11-28.
- JUAN CRUZ VIOTTI. Etcher, a powerful OS image flasher. Disponível em: <<https://etcher.io/>>. Acesso em: 2017-07-24.
- VON STUMBERG, L.; USENKO, V.; ENGEL, J.; STÜCKLER, J. ; CREMERS, D. Autonomous exploration with a low-cost quadrocopter using semi-dense monocular SLAM. In: ARXIV:1609.07835, 1st., 2016. **Anais...** [S.l.: s.n.], 2016. Disponível em: <<http://adsabs.harvard.edu/abs/2016arXiv160907835V>>. Acesso em: 2016-11-28.
- WEBER, M.; LIU, L.; JONES, K.; COVINGTON, M. J.; NACHMAN, L. ; PESTI, P. On map matching of wireless positioning data: A selective look-ahead approach. In: PROCEEDINGS OF THE 18TH SIGSPATIAL INTERNATIONAL CONFERENCE ON ADVANCES IN GEOGRAPHIC INFORMATION SYSTEMS, 18th., GIS '10, 1., 2010. **Anais...** New York, NY, USA: ACM, 2010, p. 290–299. Disponível em: <<http://doi.acm.org/10.1145/1869790.1869832>>. Acesso em: 2016-11-28.
- KISMET WIRELESS. Kismet Wireless. Disponível em: <<http://www.kismetwireless.net/>>. Acesso em: 2016-11-28.

- WU, K.; XIAO, J.; YI, Y.; GAO, M. ; NI, L. M. Fila: Fine-grained indoor localization. In: 2012 PROCEEDINGS IEEE INFOCOM, 1st., 2012. **Anais...** [S.l.: s.n.], 2012, p. 2210–2218.
- Y. YONEYAMA, M. M.; SHINODA, S. A location system with bluetooth. In: PROCEEDINGS OF THE 2002 INTERNATIONAL TECHNICAL CONFERENCE ON CIRCUITS SYSTEMS, COMPUTERS AND COMMUNICATIONS, 17th., 2002. **Anais...** [S.l.]: Kaewchai Chanchaoren, 2002. Disponível em: <<http://www.dbpia.co.kr/Journal/ArticleDetail/NODE01750539>>. Acesso em: 2016-11-28.
- YU, E.; XIONG, X. ; ZHOU, X. Automating 3d wireless measurements with drones. In: PROCEEDINGS OF THE TENTH ACM INTERNATIONAL WORKSHOP ON WIRELESS NETWORK TESTBEDS, EXPERIMENTAL EVALUATION, AND CHARACTERIZATION, 16th., WINTECH '16, 1., 2016. **Anais...** New York, NY, USA: ACM, 2016, p. 65–72. Disponível em: <<http://doi.acm.org/10.1145/2980159.2980168>>. Acesso em: 2016-11-28.
- ZHANG, Z.; ZHOU, X.; ZHANG, W.; ZHANG, Y.; WANG, G.; ZHAO, B. Y. ; ZHENG, H. I am the antenna: Accurate outdoor ap location using smartphones. In: PROCEEDINGS OF THE 17TH ANNUAL INTERNATIONAL CONFERENCE ON MOBILE COMPUTING AND NETWORKING, 11th., MOBICOM '11, 1., 2011. **Anais...** New York, NY, USA: ACM, 2011, p. 109–120. Disponível em: <<http://doi.acm.org/10.1145/2030613.2030626>>. Acesso em: 2016-11-28.
- ZHU, H.; ZHONG, X.; YU, Q. ; WAN, Y. A localization algorithm for mobile wireless sensor networks. In: 2013 THIRD INTERNATIONAL CONFERENCE ON INTELLIGENT SYSTEM DESIGN AND ENGINEERING APPLICATIONS, 1st., 2013. **Anais...** [S.l.: s.n.], 2013, p. 81–85.

## 7 APÊNDICES

## APÊNDICE 1: LISTA DE COMPONENTES PARA MONTAGEM DO QUADRIRROTOR

A lista de componentes para montagem do VANT utilizado nos ensaios experimentais dessa dissertação são apresentados na tabela 7.1. Todos os itens são descritos com o seu nome, modelo, preço (em Dólares e Reais, com o câmbio médio de março de 2017) e fornecedores nacionais ou estrangeiros, utilizados em cada compra. A estimativa de cálculo do custo de frete também é apresentada.

Os parâmetros para as escolhas de cada componente foram: preço, facilidade de acesso e conhecimento prévio sobre cada item. A plataforma do VANT escolhida também se baseou em outros projetos para aeronaves de baixo custo para pesquisa científica e sensoriamento (RICARDO M. BERNARDO; ROSA, 2017).

A plataforma tem por objetivo ser customizável e reutilizável. Componentes criados para a acomodação do hardware, como ferramentas, peças 3D impressas, fitas e presilhas, não estão descritas nessa lista, mas são referenciadas no Guia de montagem do VANT (Anexo 8.3)

Loja	Produto	Qtde	Valor Unit.	Frete	Valor US\$	Valor R\$
Amazon	LHI 3 pcs 2 in 1 RC 2-8s Lipo Li-ion LiMn Li-Fe Battery Checker with Low Voltage Buzzer Alarm and LED Indicator	1	8,00	0,00	8,00	26,40
Amazon	SummitLink DJI Flamewheel F450 Prop Guard Tool-Free Quick Snap On/off Prop Guards	1	19,95	0,00	19,95	65,84
Multicopter Warehouse	DJI E310 Tuned Propulsion System for Quadcopters	1	159,00	11,97	170,97	564,20
Amazon	DJI Flame Wheel F450 Basic Quadcopter Drone Kit	1	49,00	0,00	49,00	161,70
Emlid	Navio2 board + kit de cabos + antena GPS + PowerModule	1	143,50	23,92	167,42	552,49
RobotShop	Maxbotix I2CXL-MaxSonar-EZ0 High Performance Sonar	1	44,95	0,00	44,95	148,34
RobotShop	Self-Adhesive Pure Copper Heat Sink For Raspberry Pi	1	0,99	0,00	0,99	3,27
RobotShop	Raspberry Pi Camera Module V2	1	25,00	0,00	25,00	82,50
RobotShop	Raspberry Pi 3 Computer Board	1	39,95	0,00	39,95	131,84
3DR store	433 MHz (European) Telemetry Radio Set 3DR Original	1	39,99	11,71	51,70	170,61
eBay	Bateria Lipo Multistar 5200mAh 3S 11.1V conector XT60	1	38,95	0,00	38,95	128,54
Hobbyking	Vibration Damping Ball (frete p/ Brasil)	2	1,04	3,99	6,07	20,03
Banggood	Flysky Receiver FS-iA6B (somente receptor, com frete p/ Brasil - para quem já tem um rádio transmissor Flysky)	1	9,99	1,30	11,29	37,26
Banggood	FrSky Receiver X8R 8/16CH (somente receptor, com frete p/ Brasil - para quem já tem um rádio transmissor FrSky)	1	36,99	0,00	36,99	122,07
Banggood	Flysky Radio FS-i6 + Receiver FS-iA6B (frete p/ Brasil) (Opção econômica de rádio/receptor)	1	49,99	5,00	54,99	181,47
Banggood	FrSky Radio Taranis X9D PLUS + Receiver X8R (frete p/ Brasil) (Opção robusta de rádio/receptor)	1	209,99	0,00	209,99	692,97
Amazon	Combo Special: Tenergy TB6-B Balance Charger for NiMH/NiCD/Li-PO/Li-Fe Battery Packs + Power Supply	1	57,99	0,00	57,99	191,37
Amazon	RC lipo Safty Bag/Lipo Guard Bag For Charging Large 235*65*180mm	1	6,99	0,00	6,99	23,07
<b>Total (com opção econômica de receptor - para quem já tem rádio transmissor Flysky, carregador de bateria e bolsa de transporte seguro de baterias)</b>					<b>634,24</b>	<b>2092,99</b>
<b>Total (com opção robusta de receptor - para quem já tem rádio transmissor FrSky, carregador de bateria e bolsa de transporte seguro de baterias)</b>					<b>724,92</b>	<b>2392,24</b>
<b>Total (com carregador de bateria, bolsa de transporte seguro de bateria e opção econômica de rádio/receptor)</b>					<b>791,20</b>	<b>2610,96</b>
<b>Total (com carregador de bateria, bolsa de transporte seguro de bateria e opção robusta de rádio/receptor)</b>					<b>946,20</b>	<b>3122,46</b>
<b>Cotação do dólar</b>					<b>3,30</b>	
<b>OBSERVAÇÕES</b>						
1) Os valores não incluem impostos de importação, IOF etc.;						
2) Os valores de frete são para o armazém da Shipito.com (Califórnia, EUA), site que você deve assinar para ter um endereço americano para receber itens que não são enviados para o Brasil direto pela loja;						
3) O custo de assinatura da Shipito.com é de US\$ 50,00/ano para ter direito à consolidação (juntar várias encomendas em uma única caixa para economizar no envio para o Brasil);						
4) O custo do serviço de envio da Shipito.com para o Brasil é de US\$ 2,00/pacote recebido no armazém + US\$ 5,00 pela consolidação + US\$ ~90,00 pela taxa de envio em si via USPS Priority Mail (6 a 10 dias úteis para chegar);						
5) Os produtos enviados direto para o Brasil estão com modalidade de frete econômico e podem levar até 40 dias úteis para chegar;						
6) Ao enviar um pacote para o Brasil via Shipito.com, você preenche os dados do "invoice", com as descrições e valores dos produtos, sendo que esses valores serão usados para cálculo do imposto de importação quando os produtos chegarem aqui.						

TAB. 7.1: Lista de Componentes, com seus fornecedores e preços, para a montagem do quadricóptero utilizado na pesquisa Fonte: (RICARDO M. BERNARDO; ROSA, 2017)

## APÊNDICE 2: INFORMAÇÕES DO VOO EXPERIMENTAL

### 7.2.1 MASSA DE DADOS

#### Parâmetros de identificação de Alvo: Rádio WiFi (IEEE 802.11):

- MAC: B8:5A:73:A4:E8:9E
- Frequência de operação: 2.437 GHz
- Canal: 6
- Nome (SSID): Ancora

#### Parâmetros de identificação de Alvo: Rádio Bluetooth:

- Endereço MAC: B8:5A:73:A4:E8:9D
- Foi necessário o pareamento bluetooth, realizado previamente.

TAB. 7.2: Total de radiotransmissores IEEE 802.11 detectados

Endereço MAC	Freq. (Ghz)	RSSI médio (dBm)	Canal	Nome (SSID)
C0:4A:00:F8:80:F6	2.412	-79	1	tplink_nano
<b>F4:F2:6D:FB:5A:94</b>	2.427	-76	4	igorlandia
B8:5A:73:A4:E8:9E	2.437	-58	6	Ancora
4C:5E:0C:10:38:D8	2.462	-93	11	3X_PL_3
80:2A:A8:F2:34:95	5.3	-93	60	MD_SG58



Timestamp	IEEE 802.11		Bluetooth		Posição
	RSSI	Qualidade	RSSI		
1510920098.71	-70	50/100	-21	-20.7463785, -41.2290767, alt=6.01	
1510920099.98	-70	50/100	-18	-20.7463785, -41.2290767, alt=6.01	
1510920101.25	-70	50/100	-14	-20.7463785, -41.2290767, alt=6.01	
1510920102.52	-70	50/100	-16	-20.7463785, -41.2290767, alt=6.01	
1510920103.82	-70	50/100	-13	-20.7463785, -41.2290767, alt=6.01	
1510920105.11	-70	50/100	-15	-20.7463785, -41.2290767, alt=6.01	
1510920106.38	-70	50/100	-14	-20.7463785, -41.2290767, alt=6.01	
1510920107.65	-70	50/100	-13	-20.7463785, -41.2290767, alt=6.01	
1510920108.96	-70	50/100	-13	-20.7463785, -41.2290767, alt=6.01	
1510920110.25	-70	50/100	-16	-20.7463785, -41.2290767, alt=6.01	
1510920111.53	-71	47/100	-16	-20.7463785, -41.2290767, alt=6.01	
1510920112.82	-71	47/100	-16	-20.7463785, -41.2290767, alt=6.01	
1510920114.11	-71	47/100	-20	-20.7463785, -41.2290767, alt=6.01	
1510920115.39	-71	47/100	-13	-20.7463785, -41.2290767, alt=6.01	
1510920116.69	-71	47/100	-12	-20.7463785, -41.2290767, alt=6.01	
1510920117.99	-70	50/100	-12	-20.7463785, -41.2290767, alt=6.01	
1510920119.26	-70	50/100	-15	-20.7463785, -41.2290767, alt=6.01	
1510920120.56	-70	50/100	-16	-20.7463785, -41.2290767, alt=6.01	
1510920121.85	-70	50/100	-15	-20.7463785, -41.2290767, alt=6.01	
1510920122.16			-15	-20.7463785, -41.2290767, alt=6.01	
1510920223.1	-77	31/100	-20	-20.7465695, -41.2290741, alt=7.21	
1510920224.38	-77	31/100	-24	-20.7465695, -41.2290741, alt=7.21	
1510920225.65	-77	31/100	-21	-20.7465695, -41.2290741, alt=7.21	
1510920226.93	-77	31/100	-19	-20.7465695, -41.2290741, alt=7.21	
1510920228.22	-77	31/100	-22	-20.7465695, -41.2290741, alt=7.21	
1510920229.54	-77	31/100	-21	-20.7465695, -41.2290741, alt=7.21	
1510920230.81	-71	47/100	-20	-20.7465695, -41.2290741, alt=7.21	
1510920232.1	-71	47/100	-20	-20.7465695, -41.2290741, alt=7.21	
1510920233.41	-71	47/100	-20	-20.7465695, -41.2290741, alt=7.21	
1510920234.71	-71	47/100	-22	-20.7465695, -41.2290741, alt=7.21	
1510920236.12	-71	47/100	-23	-20.7465695, -41.2290741, alt=7.21	
1510920237.42	-72	44/100	-24	-20.7465695, -41.2290741, alt=7.21	
1510920238.7	-72	44/100	-21	-20.7465695, -41.2290741, alt=7.21	
1510920239.96	-72	44/100	-23	-20.7465695, -41.2290741, alt=7.21	
1510920241.28	-72	44/100	-25	-20.7465695, -41.2290741, alt=7.21	
1510920242.58	-72	44/100	-22	-20.7465695, -41.2290741, alt=7.21	
1510920243.87	-72	44/100	-19	-20.7465695, -41.2290741, alt=7.21	
1510920245.17	-72	44/100	-19	-20.7465695, -41.2290741, alt=7.21	
1510920246.49	-72	44/100	-20	-20.7465695, -41.2290741, alt=7.21	
1510920247.76	-72	44/100	-19	-20.7465695, -41.2290741, alt=7.21	
1510920342.76	-78	29/100		-20.7465706, -41.2292116, alt=8.61	
1510920344.07	-73	42/100		-20.7465706, -41.2292116, alt=8.61	
1510920345.37	-73	42/100		-20.7465706, -41.2292116, alt=8.61	
1510920346.66	-73	42/100		-20.7465706, -41.2292116, alt=8.61	
1510920347.97	-73	42/100		-20.7465706, -41.2292116, alt=8.61	
1510920349.28	-73	42/100		-20.7465706, -41.2292116, alt=8.61	
1510920350.56	-73	42/100		-20.7465706, -41.2292116, alt=8.61	
1510920351.85	-73	42/100		-20.7465706, -41.2292116, alt=8.61	
1510920353.14	-73	42/100		-20.7465706, -41.2292116, alt=8.61	
1510920354.43	-73	42/100		-20.7465706, -41.2292116, alt=8.61	
1510920355.72	-73	42/100		-20.7465706, -41.2292116, alt=8.61	

TAB. 7.3: Massa de dados coletados no voo de 17/11/2017

1510920357.0	-73	42/100		-20.7465706, -41.2292116, alt=8.61
1510920358.29	-73	42/100		-20.7465706, -41.2292116, alt=8.61
1510920359.56	-73	42/100		-20.7465706, -41.2292116, alt=8.61
1510920360.87	-73	42/100		-20.7465706, -41.2292116, alt=8.61
1510920362.16	-73	42/100		-20.7465706, -41.2292116, alt=8.61
1510920363.45	-80	24/100		-20.7465706, -41.2292116, alt=8.61
1510920364.75	-80	24/100		-20.7465706, -41.2292116, alt=8.61
1510920366.03	-80	24/100		-20.7465706, -41.2292116, alt=8.61
1510920367.3	-80	24/100		-20.7465706, -41.2292116, alt=8.61
1510920436.5	-80	24/100	-23	-20.7465718, -41.2292091, alt=8.01
1510920437.78	-77	31/100	-25	-20.7465718, -41.2292091, alt=8.01
1510920439.07	-77	31/100	-26	-20.7465718, -41.2292091, alt=8.01
1510920440.36	-77	31/100	-24	-20.7465718, -41.2292091, alt=8.01
1510920441.65	-77	31/100	-23	-20.7465718, -41.2292091, alt=8.01
1510920442.94	-77	31/100	-26	-20.7465718, -41.2292091, alt=8.01
1510920444.25	-82	20/100	-24	-20.7465718, -41.2292091, alt=8.01
1510920445.55	-82	20/100	-26	-20.7465718, -41.2292091, alt=8.01
1510920446.84	-82	20/100	-23	-20.7465718, -41.2292091, alt=8.01
1510920448.13	-82	20/100	-20	-20.7465718, -41.2292091, alt=8.01
1510920449.42	-82	20/100	-18	-20.7465718, -41.2292091, alt=8.01
1510920450.73	-76	34/100	-19	-20.7465718, -41.2292091, alt=8.01
1510920452.02	-76	34/100	-14	-20.7465718, -41.2292091, alt=8.01
1510920453.32	-76	34/100	-15	-20.7465718, -41.2292091, alt=8.01
1510920454.63	-76	34/100	-15	-20.7465718, -41.2292091, alt=8.01
1510920455.9	-76	34/100	-14	-20.7465718, -41.2292091, alt=8.01
1510920457.17	-80	24/100	-14	-20.7465718, -41.2292091, alt=8.01
1510920458.43	-80	24/100	-17	-20.7465718, -41.2292091, alt=8.01
1510920459.72	-80	24/100	-15	-20.7465718, -41.2292091, alt=8.01
1510920461.02	-80	24/100	-15	-20.7465718, -41.2292091, alt=8.01
1510920584.49	-60	76/100	-21	-20.7463777, -41.2293444, alt=8.04
1510920585.8	-57	83/100	-15	-20.7463777, -41.2293444, alt=8.04
1510920587.1	-57	83/100	-16	-20.7463777, -41.2293444, alt=8.04
1510920588.38	-57	83/100	-16	-20.7463777, -41.2293444, alt=8.04
1510920589.66	-57	83/100	-12	-20.7463777, -41.2293444, alt=8.04
1510920590.95	-57	83/100	-12	-20.7463777, -41.2293444, alt=8.04
1510920592.23	-58	81/100	-15	-20.7463777, -41.2293444, alt=8.04
1510920593.53	-58	81/100	-15	-20.7463777, -41.2293444, alt=8.04
1510920594.82	-58	81/100	-17	-20.7463777, -41.2293444, alt=8.04
1510920596.12	-58	81/100	-13	-20.7463777, -41.2293444, alt=8.04
1510920597.4	-58	81/100	-14	-20.7463777, -41.2293444, alt=8.04
1510920598.68	-58	81/100	-14	-20.7463777, -41.2293444, alt=8.04
1510920599.97	-58	81/100	-16	-20.7463777, -41.2293444, alt=8.04
1510920601.25	-58	81/100	-14	-20.7463777, -41.2293444, alt=8.04
1510920602.53	-58	81/100	-19	-20.7463777, -41.2293444, alt=8.04
1510920603.82	-58	81/100	-16	-20.7463777, -41.2293444, alt=8.04
1510920605.14	-64	65/100	-15	-20.7463777, -41.2293444, alt=8.04
1510920606.43	-64	65/100	-13	-20.7463777, -41.2293444, alt=8.04
1510920607.72	-64	65/100	-15	-20.7463777, -41.2293444, alt=8.04
1510920609.02	-64	65/100	-16	-20.7463777, -41.2293444, alt=8.04

TAB. 7.4: Massa de dados coletados no voo de 17/11/2017

#	Comando Aplicado	Tempo (s)	Latitude	Longitude	Altitude (m)	Ang	Dist. (m)
01	DO_CHANGE_SPEED	1	0	0	0	0	0,0
02	WAYPOINT	2	-20,7463798522949	-41,229076385498	8	0,0	0,0
03	LOITER_TIME	108	-20,7463798522949	-41,229076385498	8	NaN	0,0
04	WAYPOINT	2	-20,7465705871582	-41,229076385498	8	0,0	21,2
05	LOITER_TIME	108	-20,7465705871582	-41,229076385498	8	NaN	0,0
06	WAYPOINT	2	-20,7465705871582	-41,2292098999023	8	0,0	13,9
07	LOITER_TIME	108	-20,7465705871582	-41,2292098999023	8	NaN	0,0
08	WAYPOINT	2	-20,7465705871582	-41,2293434143066	8	0,0	13,9
09	LOITER_TIME	108	-20,7465705871582	-41,2293434143066	8	NaN	0,0
10	WAYPOINT	2	-20,7463798522949	-41,2293434143066	8	0,0	21,2
11	LOITER_TIME	108	-20,7463798522949	-41,2293434143066	8	NaN	0,0
12	WAYPOINT	2	-20,7463798522949	-41,2292098999023	8	0,0	13,9
13	LOITER_TIME	1	-20,7463798522949	-41,2292098999023	8	NaN	0,0
14	WAYPOINT	2	-20,7464809417725	-41,2292098999023	8	0,0	11,2
15	LAND	0	-20,7464809417725	-41,2292098999023	20	90,0	12,0

TAB. 7.5: Massa de pontos (*waypoints*) em missão autônoma no voo de 17/11/2017

## 8 ANEXOS

## ANEXO 1: *CHECK-LIST* DE EQUIPAMENTOS

O objetivo deste documento é prover uma lista com a série de equipamentos e material de apoio que eventualmente serão necessários durante os ensaios em campo para um voo de VANT (veículo aéreo não tripulado) do tipo quadricóptero. Essa lista foi concebida através da experiência coletiva de alunos dos cursos de Mestrado e Doutorado do Instituto Militar de Engenharia (IME) com o intuito de auxiliar novos alunos no processo de organização antes de um ensaio experimental longe da instituição, onde (devido ao isolamento) muitas vezes será impossível regressar ao laboratório ou mesmo ao comércio local em busca de ferramentas esquecidas.

Esse *checklist* deve ser seguido por todos os participantes antes de partir para uma missão de voo. Mesmo que um indivíduo não seja capaz de atender a todos os requisitos, esse deve entrar em contato com parceiros empenhados no mesmo ensaio para verificar se todos os itens terão um responsável.

Sugerimos fortemente que essa lista seja revisada mais de uma vez durante os preparativos e que todos os membros do time estejam cientes das faltas e tenham refletido sobre o que utilizar em substituição aos itens ausentes.

Revisões		
Nome do Revisor	Alteração	Data
Igor Prata	Lançamento original	29/09/2017
Igor Prata	Alterações com bases em ensaios de voo	20/10/2017
Igor Prata	Alterações com bases em ensaios de voo	17/10/2017
Igor Prata	Revisões pós edição final da dissertação	27/10/2017

### 8.1.1 LISTA DE FERRAMENTAS NECESSÁRIAS PARA TESTES DE VOOS COM *DRONES*:

- Drone* (levar montado o máximo que puder)
- Hélices desmontadas (incluindo 4 Extras)
- Caixa ou sacola com todos os itens descritos abaixo.

### 8.1.2 ENERGIA

- Extensão elétrica de 10m ou mais
- 2 adaptadores de tomadas (um para do padrão USA para o ABNT e outro vice-versa)
- Power bank* usb com cabo
- Baterias Lipo extras (3S ou 4S)
- Carregador Lipo
- Carregador de laptop
- Carregador USB
- Cabo USB para o telefone
- Bateria extra de laptop
- 4 Pilhas AA extras

### 8.1.3 FERRAMENTAS

- Estanho
- Soldador
- Chave Philips
- Chave de fenda pequena
- Chave Allen (do tamanho dos parafusos do *drone*)
- Estilete
- Pinça
- Fita métrica ou trena

- Tábua de corte
- Alicates de corte
- Alicates de bico fino
- Espuma pequena para o barômetro
- Fita crepe
- Fita isolante
- Fita Silver Tape
- Fita velcro
- Fita *double-face*
- Abraçadeiras de nylon
- Lanterna (de preferência *headlamp*)
- Binóculos
- Rolo de barbante

#### 8.1.4 INFORMÁTICA

- Notebook (configurado e com o APM Planner registrado no drone)
- Pendrive
- 2 Cabos ethernet com plugue RJ45
- Roteador WiFi no padrão AC
- Adaptador usb WiFi nos padrões N e AC
- Adaptador USB Bluetooth
- Extensão USB de 20cm
- Memory card* micro SD extra
- Adaptador USB para *memory card* micro SD

#### 8.1.5 REGISTRO

- Tripé
- Câmera filmadora/fotografia

- Caderno de notas e caneta
- Lista impressa de ferramentas
- Lista impressa do *check list* operacional do dia
- Lista impressa de peças sobressalentes disponíveis

#### 8.1.6 APOIO

- Mesa dobrável
- Bancos dobráveis
- Chapéu/boné
- Óculos escuros
- Protetor solar
- Repelente de insetos
- Papel higiênico
- Garrafa de água
- Merenda
- Lona plástica para colocar no local de decolagem
- 8 Estacas de camping

#### **Licenciamento:**

Uso, reprodução e adaptações autorizados sob licença Creative Commons CC BY-SA

## ANEXO 2: PROCEDIMENTO OPERACIONAL

### **Procedimentos necessários para testes de voo com *drones*:**

O objetivo deste documento é prover uma lista com a série de procedimentos que serão necessários para a organização dos ensaios em campo para o voo de VANTs do tipo quadricóptero. Essa lista foi concebida através da experiência coletiva de muitos alunos dos cursos de Mestrado e Doutorado do Instituto Militar de Engenharia (IME) com o intuito de auxiliar novos participantes no processo de organização antes, durante e depois de um ensaio experimental em campo.

Considera-se o tempo e o conhecimento do processo grandes vantagens em experimentos onde muitos fatores aumentam a imprevisibilidade e o isolamento do local pode levar a contratempos indesejados. Esse *check list* deve ser seguido por todos os alunos antes de partir para uma missão de voo e, mesmo que um indivíduo não seja capaz de atender a todos os requisitos, esse deve entrar em contato com parceiros empenhados no mesmo ensaio para verificar se todos os itens terão um responsável.

Sugerimos fortemente que essa lista seja revisada mais de uma vez durante os preparativos e que todos os membros do time estejam cientes das faltas e tenham refletido sobre o que utilizar em substituição aos itens ausentes. Ao final do ensaio esperamos que novas experiências sejam registradas nela garantindo que futuros alunos serão ainda beneficiados.

Revisões		
Nome do Revisor	Alteração	Data
Igor Prata	Lançamento original	29/09/2017
Igor Prata	Alterações com bases em ensaios de voo	17/09/2017
Igor Prata	Alterações com bases em ensaios de voo	17/10/2017
Igor Prata	Revisões pós edição final da dissertação	27/10/2017

## 8.2.1 LISTA DE PASSOS OPERACIONAIS PARA MISSÕES COM VANT

A lista abaixo é uma sugestão de itens que devem ser seguidos:

## 8.2.2 PREPARATIVOS ANTES DO ENSAIO

- Marcar a melhor data para o ensaio com os alunos participantes
  - Garantir a presença de ao menos um veterano com experiência prévia
  - Agendar o espaço que será utilizado para o voo (quartel ou aeroclube)
  - Definir as responsabilidades de quem levará o que (ver *check list* de equipamentos)
  - Definir as caronas e o ponto de partida de cada membro
  - Agendar com o piloto de aeromodelos profissional que acompanhará o ensaio
  - Acordar o pagamento do piloto de aeromodelos profissional
  - Verificar notícias de violência na região
- Definir escopo do que será feito e quantos voos serão necessários
  - Desenhar os ensaios e compartilhar com os demais
  - Verificar se o material necessário estará disponível no dia marcado
- Montar grade de horários para voos e quando cada ensaio estará alocado
  - Não voar nas últimas horas da tarde
  - Não voar de noite (em hipótese alguma!)
  - Não voar com chuva
  - Não voar com ventos fortes
- Imprimir listas de verificação (ferramentas, operacional (essa) e de peças sobressalentes)
- Acompanhar previsão meteorológica
- Acompanhar tempestades solares (<http://www.swpc.noaa.gov/products/ace-real-time-solar-wind>)
- Revisar a montagem do *drone* (um dia antes)

- Executar testes sem as hélices
  - Testar conexão de telemetria, RC e SSH
  - Verificar se o Arducopter está carregando automaticamente
  - Carregar e balancear todas as baterias bem como verificar o estado “físico” delas
  - Verificar sentido de rotação dos rotores sem as hélices operando pelo RC
  - Verificar se o Notebook utilizado está carregado, *bootando* e com o APMPanner configurado
  - Verificar configuração do roteador WiFi (SSID e senha)
  - Verificar se todos os dispositivos estão reconhecendo e conectando no Roteador
- Levantar peças sobressalentes e escrever uma lista
  - Verificar a lista de ferramentas
  - Verificar eventuais desistências entre os participantes e o seu impacto ao plano original
    - Anotar o contato telefônico de todos os participantes
    - Se necessário, solicitar cópia do documento de todos os participantes
  - Definir onde será o almoço, considerando a distância do restaurante e o tempo necessário

### 8.2.3 PREPARATIVOS EM CAMPO

- Ao chegar no local de voo, verificar as condições do terreno e de vento
  - Buscar por cobertura ao sol e pontos de energia elétrica
  - Refletir sobre situações de riscos como áreas a serem evitadas
  - Definir o ponto de voo e forrar com lona plástica fixada por estacas caso o terreno seja de terra ou arenoso
  - Restringir o trânsito de pessoas e veículos no local de voo utilizando marcadores visuais e deixando um responsável por advertir as pessoas
  - Reunir os participantes revisando o cronograma do dia e divisão de tarefas

- Montar a mesa e as cadeiras
- Puxar extensão elétrica entre a tomada e o local da mesa
- Instalar roteador WiFi
- Montar os notebooks que serão utilizados na ordem dos voos
  - Deixar a caixa de equipamentos próxima à bancada de operação
  - Revisar o estado físico dos *drones*
  - Instalar as baterias no RC e ligar
  - Instalar as baterias do *drone* prendendo com segurança pela cinta de velcro e ligando a alimentação e telemetria do *buser* (NÃO instalar as hélices)
  - Verifique o nível de bateria de cada célula pelo *buser onboard*
  - Verifique o nível MÍNIMO CRÍTICO do *buser* (muitas vezes ele reinicia o parâmetro por inatividade do dispositivo para o *default* de fábrica), recomendado: 3,5V por célula para garantir que o VANT poderá ter energia suficiente para regressar em caso de emergência
  - Verifique se os LEDs de sinalização laterais estão instalados e funcionando
  - Revisar softwares necessários no notebook (APMPlanner) e as conexões com o VANT (SSH, TELEMETRIA, RC)
  - ATENÇÃO: Em alguns casos, a telemetria de um drone pode influenciar na de outro. Nesses casos deixe apenas um dos *drones* ligados e desligue os rádios USB dos notebooks que não serão utilizados em um voo.
  - Execute os passos de calibração do drone pelo APMPlanner
  - No APMPlanner termine o plano de voo (se ainda não estiver pronto)
  - Salve o plano de voo num arquivo
  - Carregue o plano de voo no VANT
  - Ainda sem as hélices, arme o *drone* pelo RC (*Remote Control*) e com um *thrust* mínimo verifique se o sentido de rotação dos rotores está correto.
  - Desarme o *drone* pelo RC
- Monte os equipamentos necessários para o ensaio seguindo o desenho original
- Habilitar software de captura de tela

- Deixe o *drone* sobre o ponto de decolagem
- Desligue a bateria do *drone* e instale as hélices (ATENÇÃO à ordem correta)
- Religue a bateria e aguarde o sinal sonoro avisando a carga do serviço Arducopter
- AFASTE-SE DE UM RAIO DE 10m do *DRONE* e garanta que todos respeitaram essa distância de segurança.
- Verifique se as chaves do RC estão operando corretamente pelo APMPPlanner (MANUAL, ALTHOLD, LOITER, etc)
- Verificar se existe um membro cuidando do voo pelo laptop, outro pelo RC (ou ao lado do piloto profissional), outro com o equipamento de filmagem e outro responsável pela segurança do espaço
- Decolar o veículo no modo manual e quando este estiver a 8m de altitude, ligar o modo ALTHOLD
- Verificar se o veículo não perde altitude por 2 minutos
- Ainda no ar, ativar modo LOITER
- Verificar se o veículo não perde orientação por 2 minutos
- Pousar manualmente
- Ao pousar verificar o estado das baterias pelo *buser*, trocar se necessário
- Decolar o veículo no modo manual e quando este estiver a 8m de altitude, ligar o modo automático
- checar se o plano de voo foi seguido
- Ao pousar verificar o estado das baterias pelo *buser*, trocar se necessário
- Reexecutar o procedimento de decolagem, dessa vez com o software que se deseja executar *onboard*
- Ao terminar o ensaio, salvar os arquivos de vídeo e registros *onboard* necessários em backup no notebook
- Tomar notas importantes

- Qualquer incremento útil deve ser compartilhado com os demais participantes para evoluir o processo no mesmo dia.
- Entre cada experimento verificar possíveis impactos no cronograma e alternativas, se necessário
- Ao regressar, escreva relatório sobre o evento
- Compartilhe arquivos de vídeos e de imagem com os demais
- Atualize os arquivos de *check lists* com novos itens identificados

#### 8.2.4 ATITUDE EM CASO DE INCIDENTE

- a) **Mantenha a calma;**
- b) Verifique se alguém se feriu dando atenção e proteção;
- c) Localize o local de pouso e se aproxime com cuidado;
- d) Não se aproxime do veículo até suas hélices pararem;
- e) Tente desligar os rotores pelo RC senão desligue as baterias com cuidado;
- f) Desligue a bateria do VANT;
- g) Garanta que uma pessoa salve os dados no laptop antes de buscar o VANT;
- h) Tire fotos e vídeo do estado do VANT;
- i) Recolha o veículo e os destroços;
- j) Se ninguém precisar de ajuda médica, discuta com os colegas o que pode ter acontecido de errado para evitar um novo acidente no ensaio seguinte.

#### **Licenciamento:**

Uso, reprodução e adaptações autorizados sob licença Creative Commons CC BY-SA

## ANEXO 3: GUIA DE MONTAGEM DO VANT

Esse Tutorial visa a especificação da montagem de um drone de asa rotativa do tipo quadricóptero apropriado para as necessidades de pesquisa do Laboratório de Robótica do Instituto Militar de Engenharia (IME) com o intuito de diminuir os esforços necessários para a concepção de uma plataforma de voo genérica, preservando sua aplicabilidade, baixo custo e reprodutibilidade em diversos projetos.

Diversos problemas que demandam sensoriamento de regiões remotas e o consequente processamento dos dados sensoreados são passíveis de serem trabalhados com o uso de veículos aéreos não tripulados (VANTs). Entretanto, as plataformas prontas para uso disponíveis atualmente possuem custo elevado de aquisição e manutenção, além de terem pouca flexibilidade para modificações. Para superar esta questão, este artigo apresenta o processo de concepção detalhado de uma plataforma de VANT robusta, extensível e de baixo custo, construída com componentes “de prateleira” facilmente substituíveis, incluindo sensores básicos e com autonomia de voo suficiente para uso em diversas aplicações acadêmicas.

Todos os componentes descritos nesse guia tem os seus custos e fornecedores especificados no apêndice 7.1. A escolha de peças bem como o procedimento de montagem foi baseado no artigo (RICARDO M. BERNARDO; ROSA, 2017) e em diversos trabalhos conduzidos nas pesquisas de mestrado e doutorado do Laboratório de Inteligência Computacional do Instituto Militar de Engenharia.

Esse material foi criado graças ao esforço de diversos alunos que participaram das cadeiras de mestrado do Instituto Militar de Engenharia bem como diversos tutoriais disponibilizados publicamente, que levaram ao sucesso da montagem dessa plataforma de pesquisa. O artigo “Concepção de uma Plataforma de VANT de Baixo Custo do Tipo Quadricóptero para Uso em Pesquisas” apresenta um detalhamento formal das vantagens de utilização dessa plataforma de voo para a comunidade científica brasileira, voltada a pesquisa de sensoriamento em voos autônomos.

A plataforma de voo é de baixo custo e é altamente customizável, alterações podem ser feitas a esse procedimento caso necessário. Arquivos e manuais auxiliares poderão ser referenciados. É necessária habilidade com estações de solda para eletrônica, ferramentas de corte e impressão de peças em 3D para o melhor aproveitamento desse material.

Revisões		
Nome do Revisor	Alteração	Data
Igor Prata	Lançamento original	29/09/2017
Igor Prata	Alterações com bases em ensaios de voo	17/09/2017
Igor Prata	Alterações com bases em ensaios de voo	17/10/2017
Igor Prata	Revisões pós edição final da dissertação	27/10/2017

### 8.3.1 INVENTÁRIO

Primeiramente serão listados todos os itens necessários para a montagem da plataforma de voo concebida nesse trabalho. A próxima seção exibe a lista de componentes, incluindo os principais, imprescindíveis para a construção do VANT, as peças customizadas, necessárias para a acomodação da eletrônica embarcada, e as ferramentas e componentes secundários, destinados a facilitar a montagem da plataforma.

### 8.3.2 EQUIPAMENTOS PRINCIPAIS

O Anexo 8.1 exibe uma lista detalhada de peças e preços estimados. A maior parte dos fornecedores são de países estrangeiros e o custo do frete e impostos deve ser considerado no momento da aquisição.

### 8.3.3 MODELAGEM DE PEÇAS 3D

Além da listagem de equipamentos que devem ser adquiridos, algumas peças customizadas devem ser fabricadas, pelo próprio usuário, para a correta montagem na estrutura principal do VANT. Componentes como os sistemas computacionais embarcados, sensores e até presilhas podem ser construídos em impressão 3D com plástico ABS ou PET através dos modelos previamente desenvolvidos presentes no seguinte arquivo (levantado na plataforma Redmine do RoboIME):

<http://redmine.robome.com.br/attachments/download/388/>

A listagem completa de componentes já modelados em 3D pode ser vista abaixo. Eventuais peças podem ser desenvolvidas em condições específicas, necessárias à pesquisa:

- a) F450-CaseNavio2-CascoInferior.stl
- b) F450-CaseNavio2-CascoSuperior.stl
- c) F450-CaseNavio2-FrameFixComGPS.stl

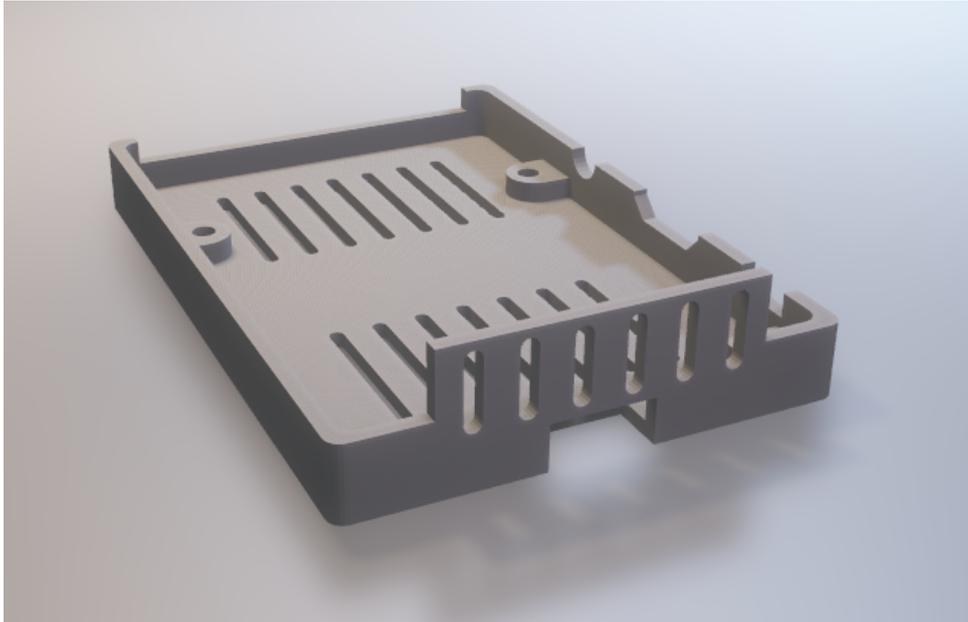


FIG. 8.1: Exemplo de componente 3D presente no arquivo de peças para impressão: Casco Inferior do sistema computacional embarcado

- d) F450-CaseNavio2-FrameFixSemGPS.stl
- e) F450-CaseNavio2-NavioToFrameFix.stl
- f) F450-CaseNavio2-SuporteGPSHaste60mm.stl
- g) F450-CaseNavio2-SuporteGPSTopo.stl
- h) F450-Gimbal-Base.stl
- i) F450-Gimbal-CameraFix.stl
- j) F450-Gimbal-FrameFix.stl
- k) F450-Gimbal-HasteBase.stl
- l) F450-Gimbal-RollArm.stl
- m) F450-Gimbal-ServoCase.stl
- n) F450-GrampoPrendedorESCE310.stl
- o) F450-LandingGear50mm.stl
- p) F450-SuporteGavetaRadioTelemetria3DR.stl
- q) F450-SuporteGavetaReceptorRadioX8R.stl

- r) F450-SuporteMesaTrabalho.stl
- s) F450-SuporteSonarComAmortecedor-Base.stl
- t) F450-SuporteSonarComAmortecedor-Corpo.stl
- u) F450-Gimbal-HasteBase NEW.stl
- v) F450-TampaLateral.stl
- w) Lidar\_laser\_holder new.STL

#### 8.3.4 ITENS EXTRAS E FERRAMENTAS

Para a correta montagem dos componentes, algumas ferramentas serão necessárias bem como certos componentes auxiliares:

- a) Ferro de solda
- b) Chave de fenda pequena
- c) Chave de Allen 2mm
- d) Estilete ou canivete de corte (para acabamentos necessários)
- e) Lima chata pequena (para acabamentos necessários)
- f) Lima cilíndrica pequena (para acabamentos necessários)
- g) 10 parafusos de 2,5 x 12 mm
- h) 2 porcas de 2,5 mm
- i) 15 borrachas para *Gimbal* de Diâmetro: 12mm/6mm e Altura: 14mm
- j) 20cm de Velcro *double-face*
- k) 20cm de Velcro com face adesiva
- l) Fita Isolante líquida
- m) Fios com plugues
- n) 10 Braçadeiras de nylon de 3mm de espessura
- o) Cabo de 10cm com plugue XT60 macho

p) Cartão de Memória MicroSD (mínimo 8GB de espaço)

Para a montagem da estrutura de câmera com amortecimento inteligente (*Gimbal*), além de alguns componentes 3D já listados anteriormente (no arquivo com os modelos) é imprescindível a aquisição dos seguintes itens:

a) Ubec de 5V 6V e 3A

b) 2 motores Micro Servo Motor Tower Pro Mg90s Metal Gear

### 8.3.5 PASSOS DE MONTAGEM

Esse capítulo exhibe os passos de montagem necessários para o *assemble* de todos os itens, apresentados nas sessões anteriores, bem como a correta configuração da plataforma de voo. A primeira parte apresenta as peças 3D já impressas e preparadas para a montagem e as seções seguintes exibem as etapas de construção com imagens ilustrativas e links para vídeos para facilitar a instrução ao usuário.

### 8.3.6 APRESENTAÇÃO DOS COMPONENTES

Abaixo são exibidos os itens adquiridos para a montagem do VANT sugerida por esse tutorial:

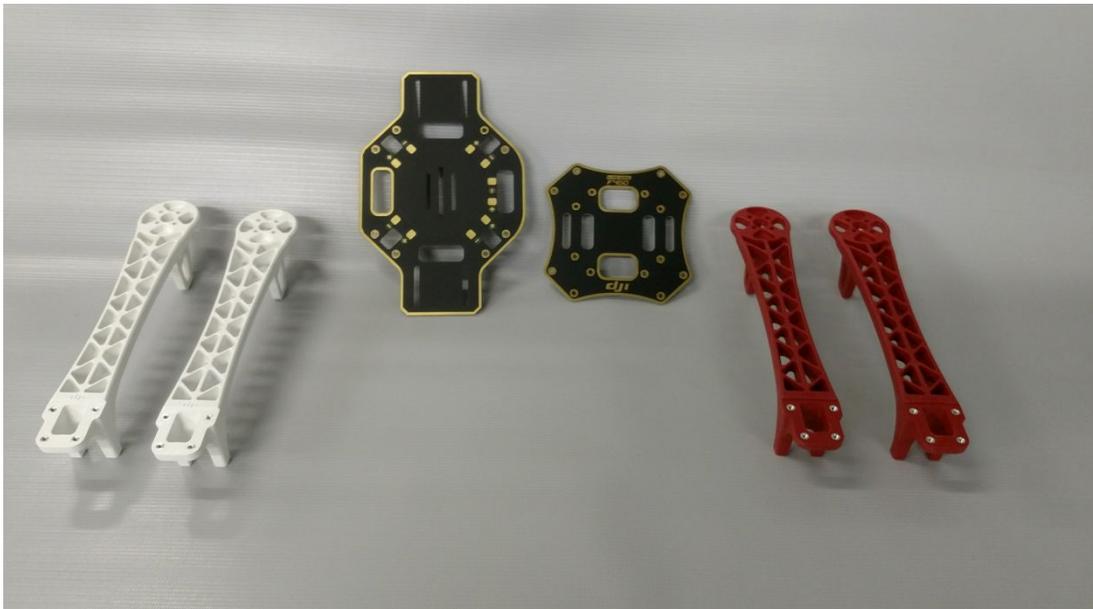


FIG. 8.2: Itens da estrutura principal do *frame* DJI modelo F450



FIG. 8.3: Kit de propulsores, hélice e ESCs DJI modelo 4205



FIG. 8.4: Protetores de Hélices para o *frame* DJI F450

### 8.3.7 PASSOS PARA CONSTRUÇÃO

Nessa seção serão exibidos os passos para a construção do VANT apresentando os procedimentos de assemble dos componentes “físicos” e em seguida a configuração “lógica” dos softwares embarcados e na estação de solo.

#### **Hardware:**

Para a montagem dos componentes de Hardware deve-se começar pela soldagem dos cabos dos ESCs e dos cabos ligados ao plugue XT60 macho.

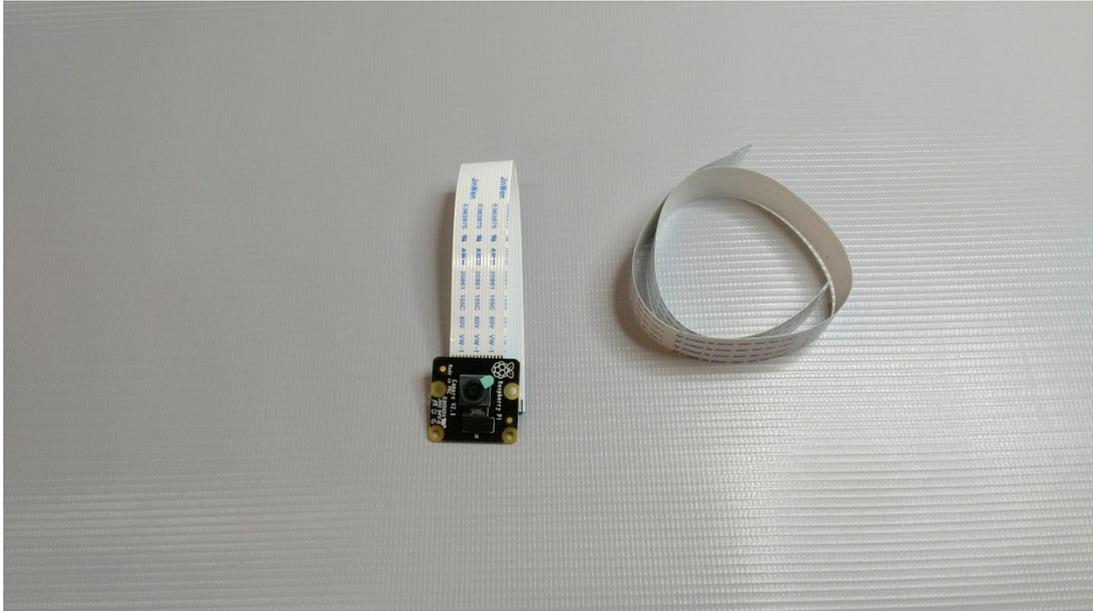


FIG. 8.5: Módulo de câmera Raspberry Pi v.2 e cabos *flat* de tamanhos diferentes



FIG. 8.6: Placa de piloto automático NAVIO2 da Emlid

A solda deve ser aplicada respeitando a polaridade descrita na placa de distribuição do *frame* DJI F450 Positivo (+) para os cabos vermelhos e Negativos (-) para os cabos negros (Figura 8.10). A solda deve ser forte o suficiente pois o veículo está sujeito a sofrer constante trepidação e colisões. Repita o procedimento com os 4 ESCs, desencapando os cabos e expondo 4 mm de metal para melhorar o contato.

Opcionalmente (caso seja necessária a montagem do *gimbal* da câmera) devem ser soldados os cabos de entrada do UBEC de 5V nos terminais de entrada.

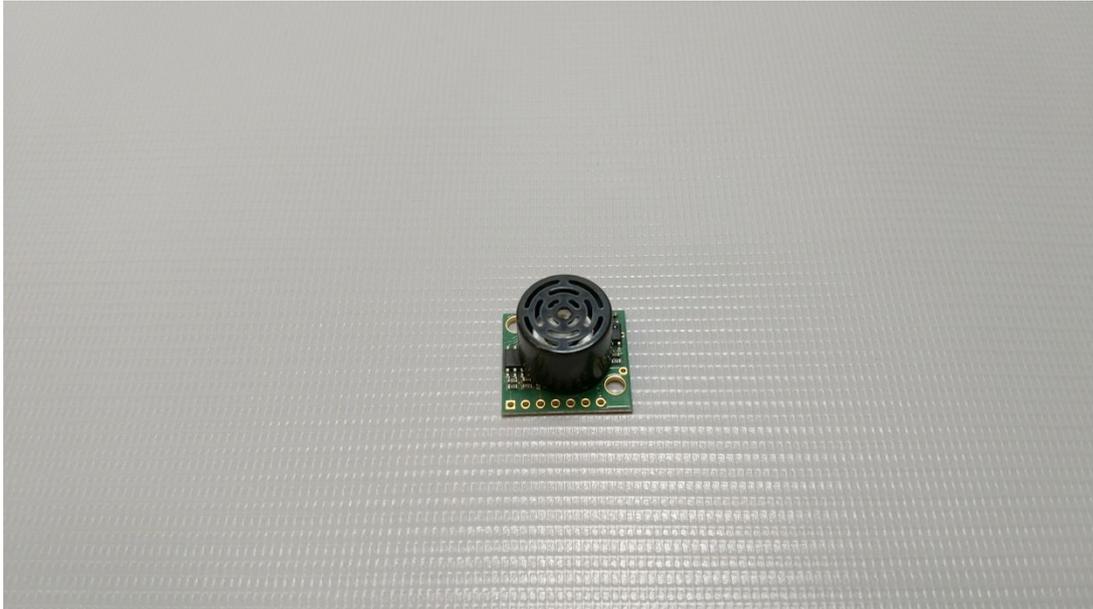


FIG. 8.7: Sonar Maxbotix modelo EZ0 Digital

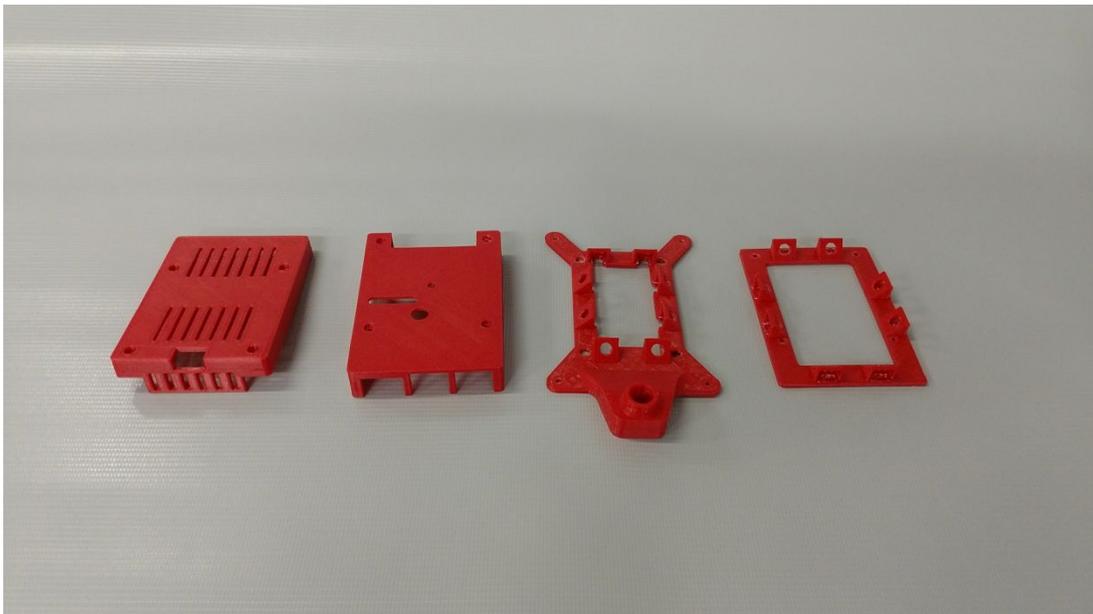


FIG. 8.8: Casco customizado para montagem do sistema computacional embarcado. Impressão em plástico PLA

Após o processo de solda, deve ser realizado o isolamento elétrico dos contatos com a fita isolante líquida de modo que ela cubra toda a solda e base dourada da placa condutora, como exibido na figura 8.11.

Utilizando a chave Allen de 2mm e os parafusos fornecidos pela DJI, montar a estrutura do frame F450, começando pela parte de cima e depois pela parte de baixo com a placa de distribuição já com os componentes soldados, figura 8.12. Serão utilizados 16 parafusos para a fixação da placas e braços do VANT. A placa superior já deve ser



FIG. 8.9: Kit de rádios para telemetria 3DR 433MHz

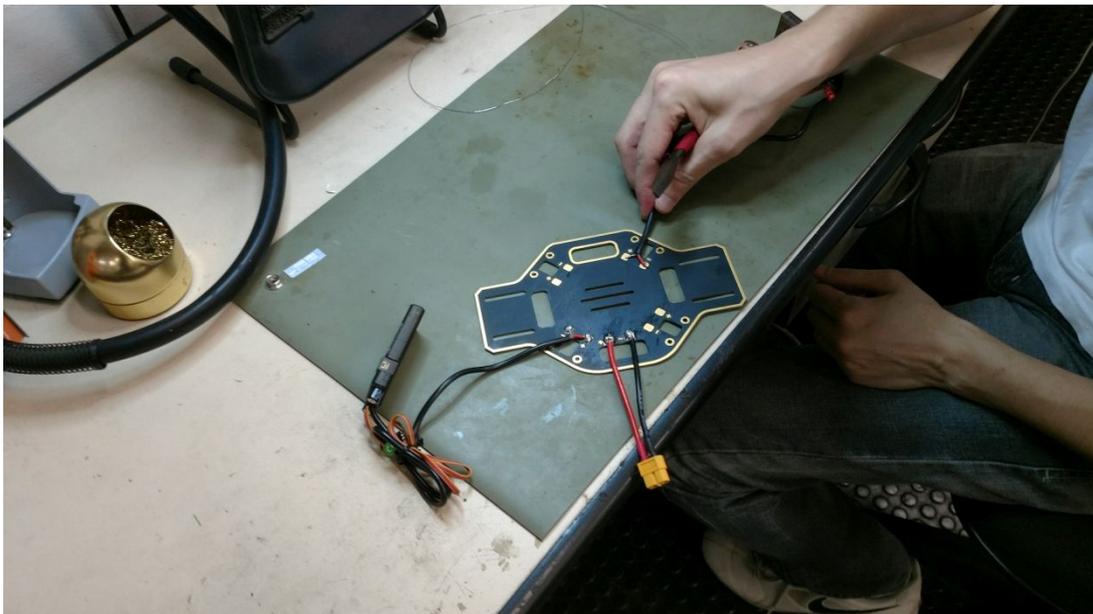


FIG. 8.10: Processo de soldagem da placa de distribuição elétrica do *frame* F450, ESCs e conector de alimentação da bateria

aparafusada com a base impressa em ABS do sistema embarcado.

Os rotores devem ser fixados nas extremidades dos braços do *frame*. Cada rotor utiliza 4 parafusos em sua base que devem estar bem apertados antes da fixação dos trens de pouso (figura 8.14). Utilize as braçadeiras de nylon para prender os ESCs na estrutura do *frame* a uns 5 cm de distância de cada rotor. Utilize 2 braçadeiras nas extremidades de cada ESC, como exibido na figura 8.13.

Como exibido na figura 8.14, os 3 cabos devem passar para a parte de baixo do braço

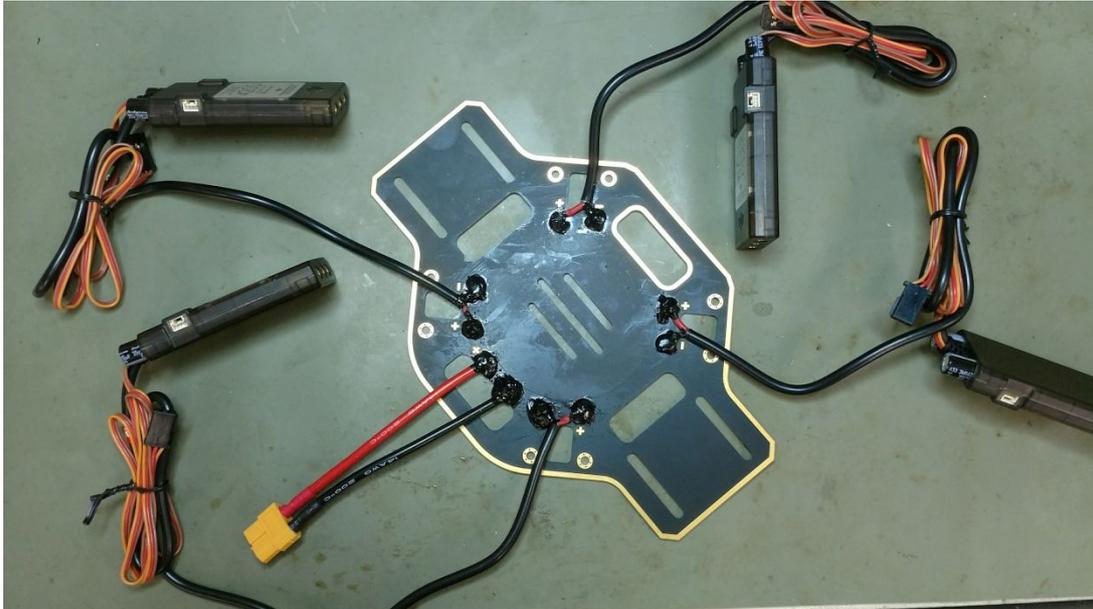


FIG. 8.11: Placa de distribuição com todos os terminais já soldados e isolados com o auxílio de pasta isolante elétrica

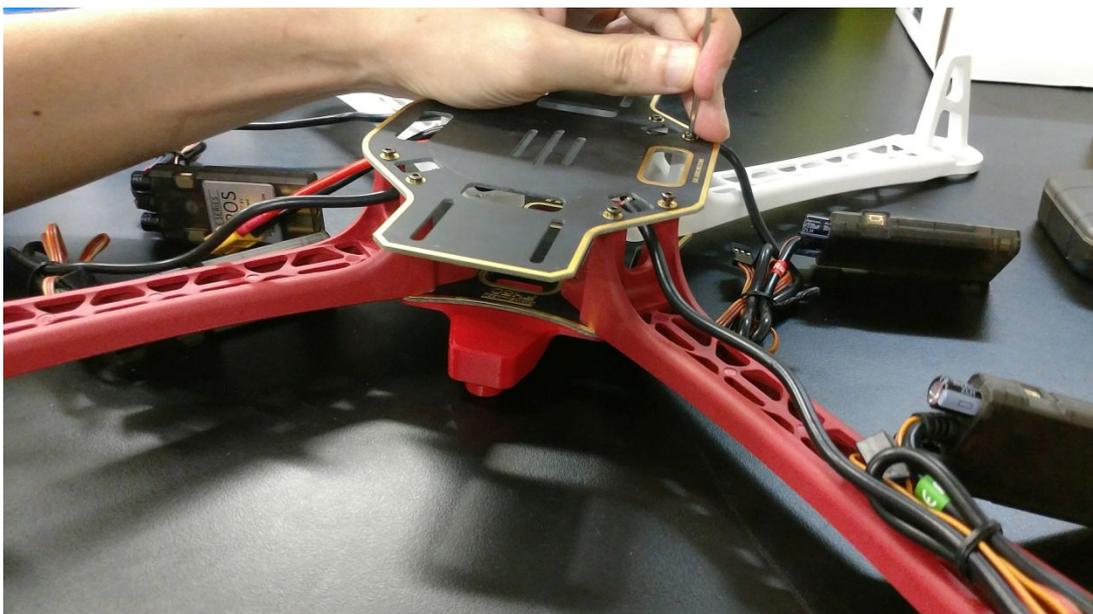


FIG. 8.12: Processo de montagem dos raios do *frame* F450, fixando à placa distribuidora de energia

aproveitando os próprios orifícios da estrutura. Atenção: deve-se sempre tentar manter o peso equalizado, para isso use os mesmos orifícios para os cabos nos 4 braços da estrutura, o mesmo vale para medir a posição de cada ESC.

Uma vez que os principais componentes foram fixados, pode-se encaixar os trens de pouso impressos em 3D no engate abaixo de cada rotor. Se necessário utilize alguma cinta de nylon para melhorar a firmeza do engate. Virando o VANT de cabeça para



FIG. 8.13: Fixação do ESC com braçadeiras de nylon



FIG. 8.14: Fixação dos rotores por parafusos e passagem dos cabos dos rotores pela estrutura do *frame*

baixo, procure algum contato ou ponto de solda ainda exposto por baixo e preencha com mais fita isolante líquida até não ser mais visível, figura 8.15.

Alguns componentes impressos podem contar arestas, farpas e estruturas adicionais utilizadas na construção. Remova esses excessos com a ajuda de limas, brocas ou estilete para prover um bom acabamento e principalmente para não forçar os componentes eletrônicos durante o encaixe. Opcionalmente pode-se utilizar um torno para melhor finalização, figura 8.16.

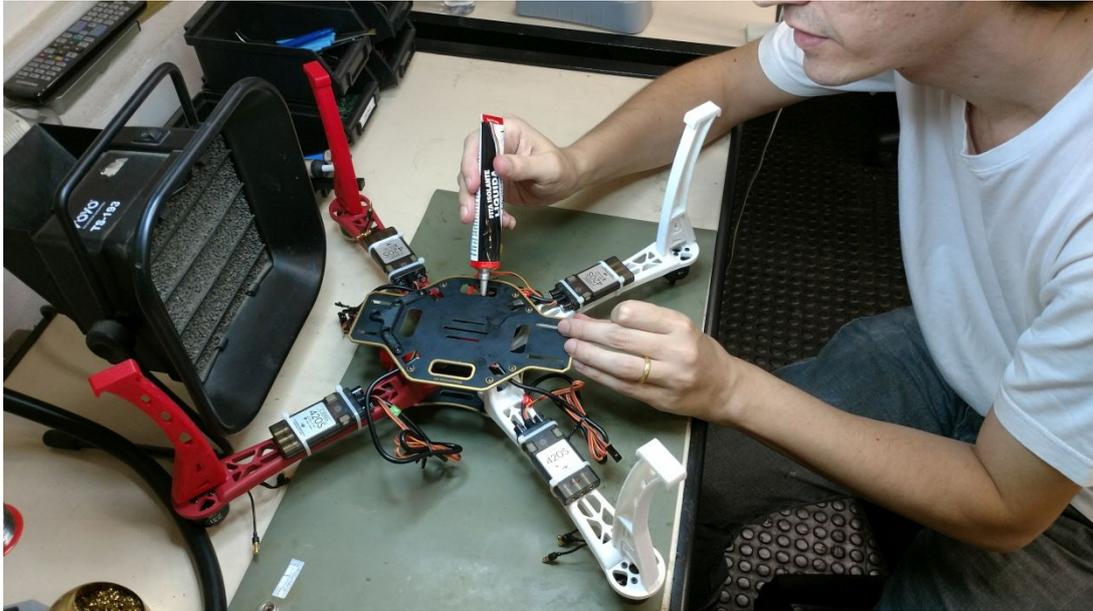


FIG. 8.15: Acabamento com fita isolante líquida nos pontos de contato ainda expostos



FIG. 8.16: Acabamento sendo aplicado em peças em plástico que contém imperfeições. Uma broca de precisão foi utilizada em algumas partes para nivelar os furos para o encaixe da placa Raspberry Pi

Após esperar o tempo de secagem da fita isolante líquida, monte os cascos superior e inferior para os computadores de bordo instalando 8 borrachas de *gimbal* entre as bases para evitar trepidações, figura 8.17.

A base superior deve ser fixada ao case montado já com as placas Raspberry Pi 3, NAVIO2, cabeçote de extensão e espaçadores por dentro utilizando 4 parafusos de 2,5 x 12mm por baixo - Seguindo a ordem apresentada na figura 8.8:

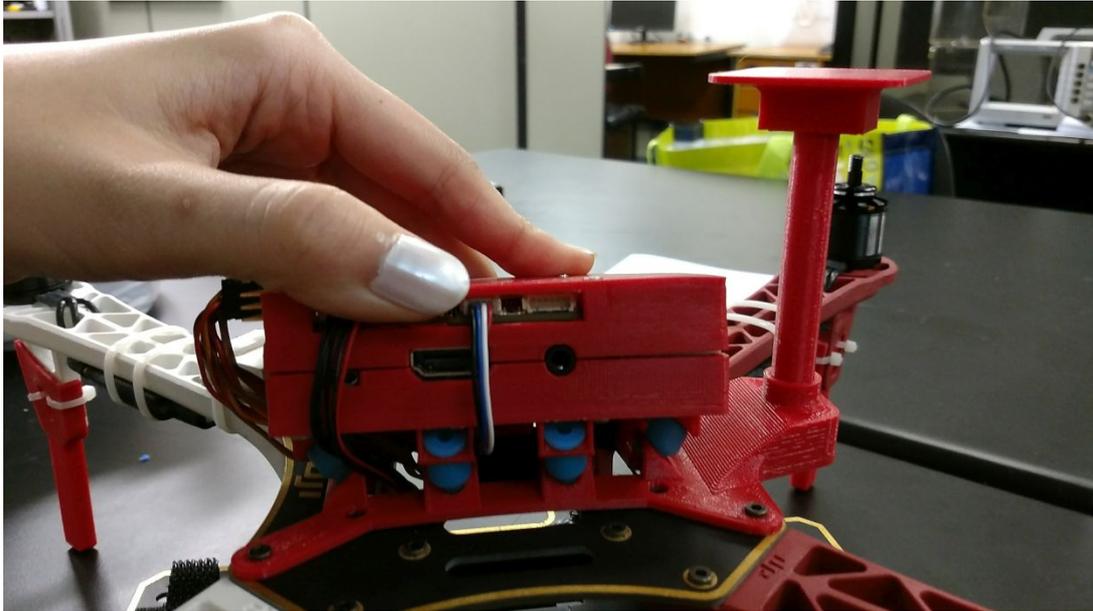


FIG. 8.17: Montagem final dos cascos e estabilizadores em borracha

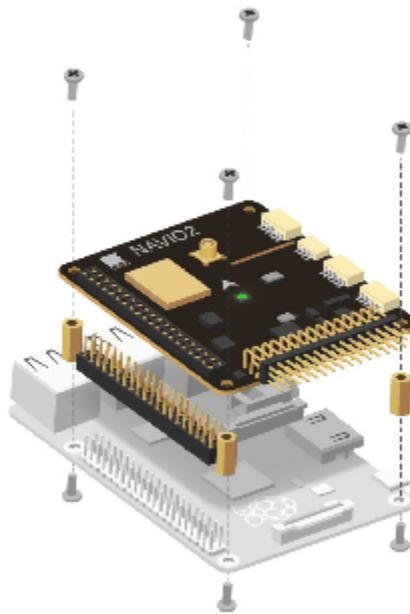


FIG. 8.18: Ordem de montagem das placas que constituem o computador de bordo do VANT (Fonte: Emlid)

Ao final, a tampa do casco superior deve ser fechada com mais 4 parafusos de 2,5 x 12mm por cima, eles encaixaram nos orifícios da NAVIO2 e nos espaçadores.

Plugue o cabo de alimentação de energia na NAVIO2 como exibido no diagrama da figura 8.20. Esse circuito deve ser ligado entre a bateria e o plugue XT60 macho soldado previamente na placa de distribuição do *frame*:

Passes os cabos de comando dos ESCs por baixo da estrutura montada e plugue no

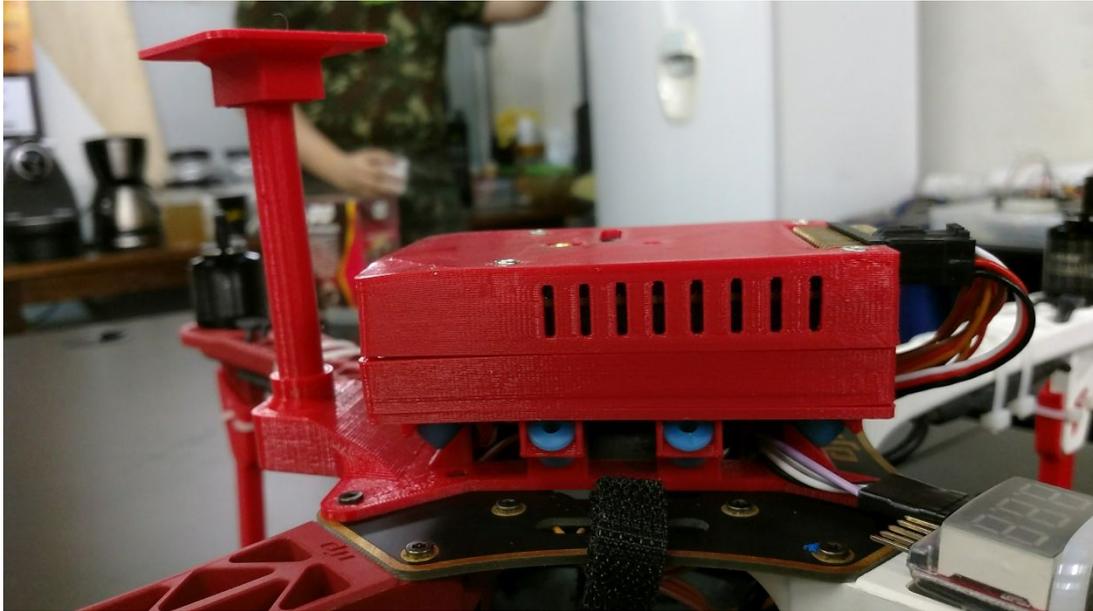


FIG. 8.19: Montagem final do casco de proteção do computador de bordo da aeronave

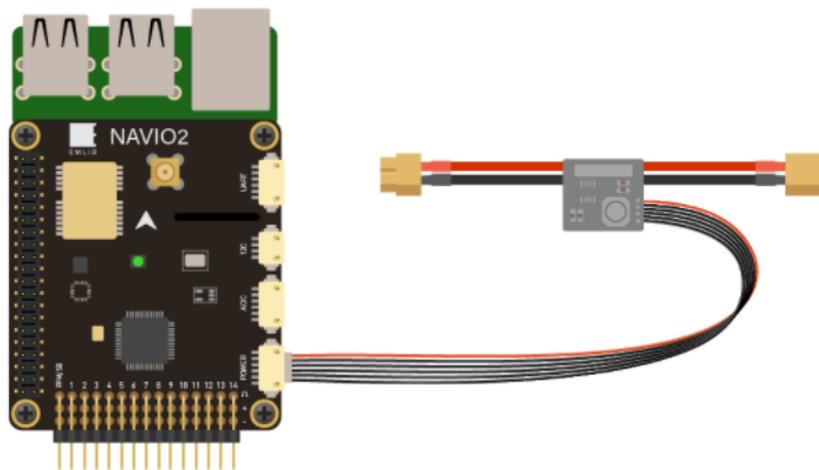


FIG. 8.20: Diagrama da placa Navio2 com destaque para a conexão com o modulo de potência (Fonte: Emlid)

barramento do Navio respeitando os contatos descritos no manual de montagem na Emlid (EMLID, 2016). Com uma braçadeira de nylon fixe o *Buzer* de verificação da bateria em um dos braços do *frame*.

Conecte o terminal do receptor do rádio (RC) PPM na porta 0 (zero) na parte de trás da placa Navio2 (exposta para fora do casco) e cada um dos quatro ESCs montados ligados às portas 1, 2, 3 e 4 da mesma placa. o local correto para a conexão pode ser consultado na figura 8.22.

A ordem de cabeamento de todos os periféricos na placa NAVIO2 pode ser observada

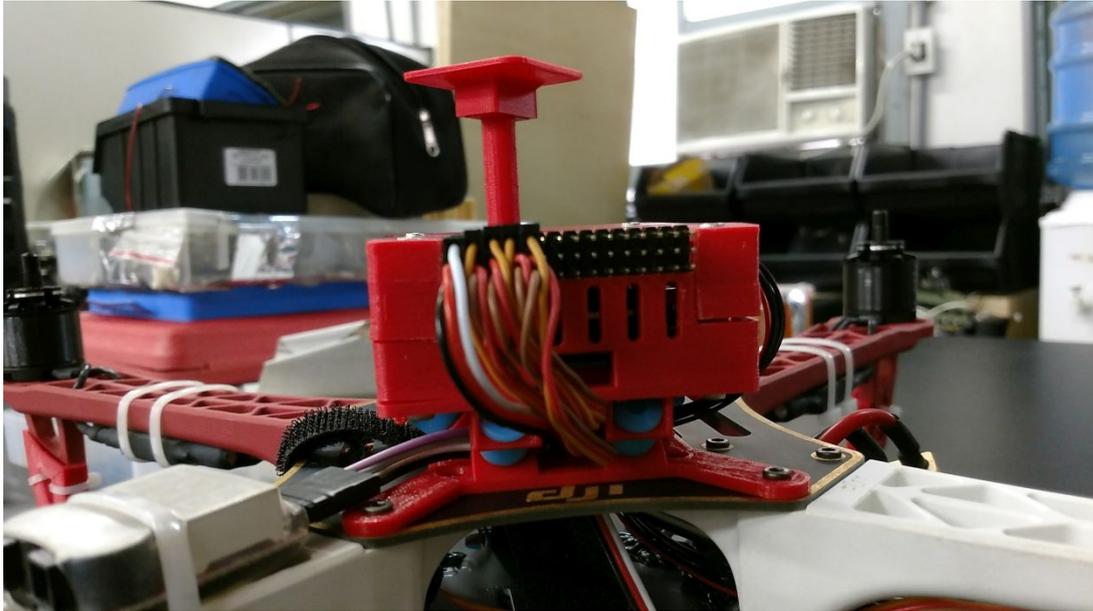


FIG. 8.21: Parte traseira do computador de bordo com os terminais PPM e PWM exposto para conexão com o receptor de rádio e ESCs dos motores

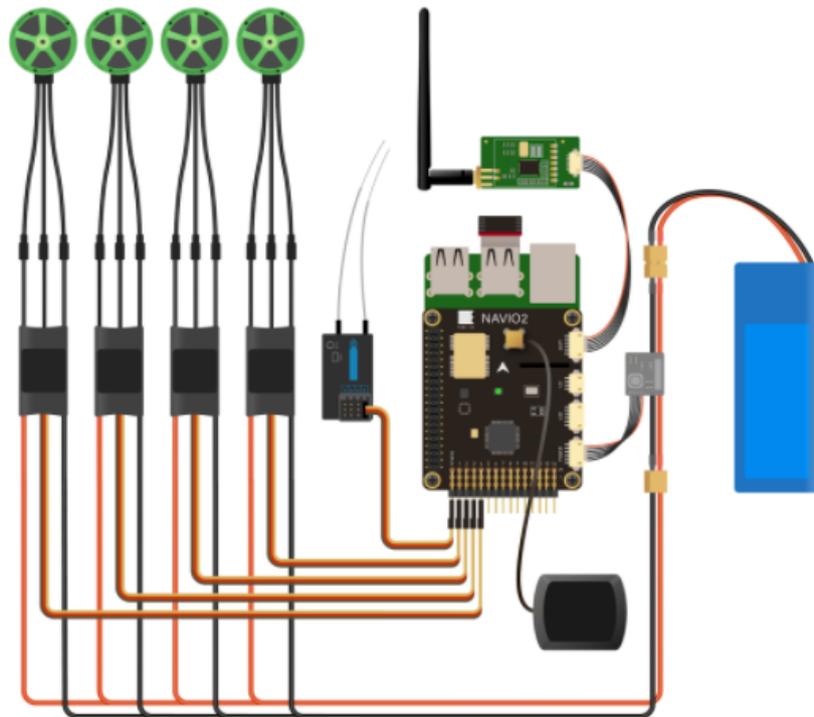


FIG. 8.22: Ordem de pinagem dos conectores PWM na Navio2 (Fonte: Emlid)

na figura 8.23 com os componentes mais necessários para o VANT.

Com a conclusão desses passos, a estrutura principal foi montada. Por motivo de segurança, para as etapas de configuração do *software* deve-se retirar as hélices.



FIG. 8.23: Diagrama de conexão de periféricos na placa Navio2

### 8.3.8 MONTAGEM DO SONAR

Para montar o sonar (importante para o piloto automático controlar o processo de pouso automatizado), primeiramente deve-se soldar 4 contatos na placa da Matbotix EZ0 para facilitar o cabeamento, só então encaixar e aparafusar no case impresso (e já com o acabamento feito) previamente. Utilize 2 parafusos de 2,5mm x 12mm e duas porcas de 2,5mm (figura 8.24). Fixe com parafusos a base do sonar sob a placa de distribuição do *frame* F450, utilize 3 borrachas para *gimbal* e conecte a base e o case do sonar por eles, garantindo estabilização contra trepidações.

Atenção à ordem de cabeamento entre a placa do sonar e o conector I2C da placa NAVIO2, como exibido na figura 8.25.

### 8.3.9 CONCLUSÕES SOBRE O PROCESSO DE MONTAGEM

Esse documento apresentou os passos necessários para a montagem de um VANT quadricóptero para fins de pesquisa utilizando componentes de fácil acesso no mercado, de baixo custo e de simples reprodução para uso pela comunidade científica. *Drones* desse tipo podem facilitar pesquisas na área de sensoriamento e navegação autônoma e são altamente customizáveis para atender uma miríade de objetivos.

A plataforma de voo descrita nesse documento já está em uso em diversos projetos no IME e este modelo se tornou o padrão base de construção de VANTs para dinamizar futuras pesquisas no laboratório de Inteligência Computacional do instituto.

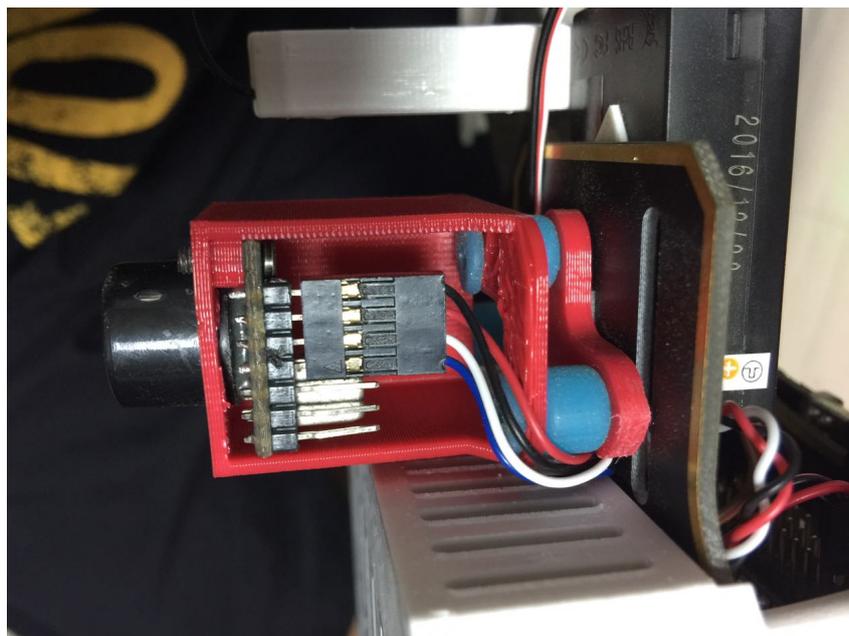


FIG. 8.24: Fixação do sonar Maxbotix EZ0 no case, impresso em plástico PLA, na base do *frame* F450 - foram utilizadas borrachas de amortecimento

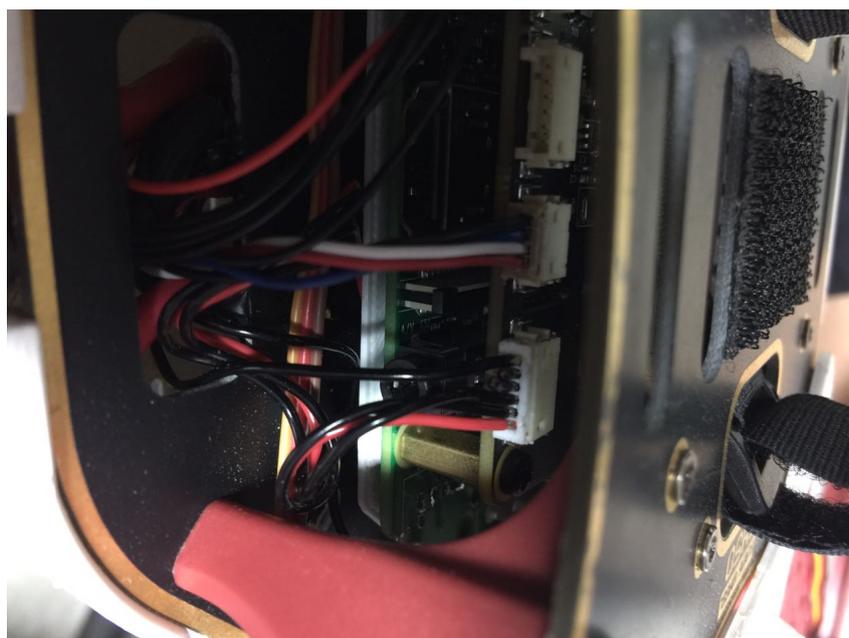


FIG. 8.25: Cabeamento I2C na placa Navio2. A ordem dos cabos deve ser seguida ao soldá-lo no sonar

Este tutorial será estendido em breve com um vídeo instrutivo com o passo-a-passo da montagem para melhor ilustrar o processo aos interessados.

**Licenciamento:**

Uso, reprodução e adaptações autorizados sob a licença Creative Commons CC BY-SA

## ANEXO 4: ESPECIFICAÇÃO DE *HARDWARE* EMBARCADO

Esse documento exhibe as especificações técnicas de cada dispositivo utilizado durante a fase de ensaios experimentais. Todo o *hardware* é de acesso público sendo encontrado em diferentes franquias de revendedores. Esses equipamentos não sofreram nenhuma customização elétrica ou mecânica que alterasse as suas especificações de fábrica.

Todos os itens descritos possuem documentação disponível na página de internet de seus fabricantes, onde mais detalhes podem ser consultados. Os *softwares* e protocolos de operação foram mantidos como o padrão.

O objetivo desse documento é prover maior transparência quanto a elaboração dos ensaios e facilitar o trabalho de reprodução dos experimentos.

Revisões		
Nome do Revisor	Alteração	Data
Igor Prata	Lançamento original	29/09/2017
Igor Prata	Alterações com bases em ensaios de voo	20/10/2017
Igor Prata	Alterações com bases em novas pesquisas	18/11/2017
Igor Prata	Revisões pós edição final da dissertação	03/12/2017

## 8.4.1 RÁDIOS

### 8.4.1.1 WIFI

#### **Raspberry Pi 3 Model B:**

- Form factor tags: onboard
- Barramento: SDIO
- Chipset: Broadcom BCM43438
- Protocolos: (802.11) bgn
- MIMO config: 1x1:1
- Conector de antena: N/A
- Observações: Antena soldada ao PCB
- FCC ID: 2ABCB-RPI32

#### **TP-LINK Archer T2UH:**

- Form factor tags: dongle
- Conector: USB 2.0 A (macho)
- Chipset: MediaTek MT7610U
- Protocolos: (802.11) abgn+ac
- MIMO config: 1x1:1
- Conector de antena: RP-SMA
- Observações: Uma antena externa destacável de ganho de 3dBi. 802.11ac (433 Mbps) 802.11n 2.4GHz (150 Mbps)
- FCC ID: TE7T2UH

#### 8.4.1.2 BLUETOOTH

##### **Raspberry Pi 3 Model B:**

- Form factor tags: onboard
- Barramento: SDIO
- Chipset: Broadcom BCM43438
- Protocolos: 4.1 (BLE + EDR)
- Conector de antena: N/A
- Observações: DESABILITADO pelo *Kernel* do sistema operacional. Antena soldada ao PCB
- FCC ID: 2ABCB-RPI32

##### **Adaptador IOGEAR**

- Form factor tags: dongle
- Conector: USB 2.0 A (macho)
- Chipset: BCM20702A0
- Frequência: 2.4GHz - 2.4835GHz (ISM Band)
- Protocolos: 4.0 (BLE + EDR + A2DP)
- Conector de antena: N/A
- Observações: banda de até 3.0 Mbps, alcance de transmissão 10 m
- Potência de Transmissão: 4 dBm (Class 2),
- Sensitivity <-82 dBm at < 0.1% BER
- Modulação: GFSK for 1 Mbps, 4-DQPSK for 2 Mbps, 8-DPSK for 3 Mbps

### 8.4.1.3 COMANDO E TELEMETRIA

#### **Kit 3D Robotics 433MHZ:**

- Potência máxima de saída: 100 mW (ajustável)
- receive sensitivity: -117 dBm
- Barramento: interface UART
- chipset: Si1000
- conector de antena: RP-SMA
- 2-way full-duplex communication through adaptive TDM
- Protocolo: MAVLink framing
- Frequency Hopping Spread Spectrum (FHSS)
- Observações: Configurable duty cycle. Error correction corrects up to 25% of bit errors. Open-source SIK firmware. Configurable through GCS.
- Alimentação: 3.7-6 V (conexão USB ou DF13). Transmit current: 100 mA at 30 dBm. Receive current: 25 mA. Interface Serial: 3.3 V

#### **Receptor FlySky FS-IA6B:**

- Número de Canais: 8
- Faixa de frequência: 2.4055 até 2.475GHZ
- Largura de banda: 140
- Protocolo: PPM
- Potência de transmissão: < 20dBm
- Sensibilidade do receptor de RF: -105dbm
- Codificação: GFSK
- Comprimento da antena: 26mm\*2 (duas antenas)
- Potência de entrada: 4.0 até 6.5 V

## 8.4.2 ALIMENTAÇÃO

### **Baterias:**

- Tipo: LiPO
- Células: 4S
- Capacidade: 5200mAh
- Tensão: 14.8V
- Constant Discharge: 10C

### **Circuito de potência da Navio2:**

- Suporte: Baterias LiPo até 6S
- Corrente máxima suportada: 60A
- Switching regulator output: 5.3V 2.25A
- Conector: Cabo DF13 de 6 terminais para conexão com o Navio2 power port
- Observações: Monitora circuitos com corrente de até 90A mas necessita de um conector diferente uma vez que o XT60 suporta apenas 60A de corrente.

### **Raspberry Pi Official Universal Power Supply Unit:**

- Tensão: 5V
- Corrente Máxima de saída: 2A

## 8.4.3 SENSORES DE BORDO

### **Sonar MAXBOTIX:**

- modelo: Maxbotix Sonar EZ0 (I2CXL)
- Serie: 1202
- Alimentação: 3V ou 5.5V
- Frequência de leitura: 5mS (40Hz) para objetos próximos e 15Hz para objetos no alcance máximo

- Ultrasonic signal frequency: 42KHz
- Barramento: I2C

#### **Lidar Lite V3 Garmin:**

- Resolução: 1 cm
- Precisão: +/- 2,5 cm a distâncias maiores que 1 m.
- Alcance: 0-40 m
- Taxa de atualização: até 500 Hz
- Interface: I2C ou PWM
- Energia (tensão de operação): 4,75-5,5 VCC; 6 V máx.
- Consumo de corrente: <2ma ocioso; <100 mA em operação contínua
- Temperatura de operação: -20° a 60° C
- Comprimento de onda do laser/Pico de energia: 905 nm/1,3 watts
- Divergência do feixe: 4 m radianos x 2 m radianos
- Abertura óptica: 12,5 mm
- Regulamentação: Produto Laser CLASSE 1 Classificado por EN/IEC 60825-1 2007.

#### 8.4.4 DISPOSITIVO ALVO

##### ***Smartphone Samsung Galaxy Win 2:***

- Sistema Operacional: Android 5.0.2
- Memória RAM: 1 GB de RAM
- Memória Interna: 8 GB de armazenamento interno
- Chipset: Qualcomm MSM8916 Snapdragon 410
- CPU: Quad-core 1.2 GHz Cortex-A53
- GPU: Adreno 306

- Bluetooth: 4.0, A2DP
- WLAN: WiFi 802.11 b/g/n, Wi-Fi Direct, hotspot
- GNSS: GPS, with A-GPS, GLONASS, BDS
- Capacidade da Bateria 2.000 mAh

**Licenciamento:**

Uso, reprodução e adaptações autorizados sob licença Creative Commons CC BY-SA

## ANEXO 5: PROCEDIMENTO DE INSTALAÇÃO E CONFIGURAÇÃO DO SIMULADOR DE VOO

### **Passos para configuração do Ambiente de Simulação utilizando o sistema de *Software in the loop* do *framework* Dronekit:**

Esse procedimento visa a atender a necessidade de documentação dos processos de instalação, configuração e execução apropriada de um ambiente de simulação para veículos não tripulados que utilizem o Ardupilot como subsistema de piloto automático.

O sistema computacional utilizado na simulação deve ter acesso à internet para instalação dos pacotes e baixar mapas das missões de voo. O serviço de simulação se comportará como uma instância do Ardupilot, podendo receber conexões, possibilitando a execução dos programas embarcados, como em um *drone*. Sugerimos apenas atenção as diferenças no tempo de execução entre o ambiente simulado e o computador de bordo da aeronave, bem como a diferença de arquitetura do processador ao compilar os programas.

A simulação de ensaios para validação de missões de voo e execução de código antes de um voo com um VANT é mandatório. Experimentos dessa natureza ajudam na previsão de problemas que podem ser encontrados em campo e mitigam diversos riscos.

Revisões		
Nome do Revisor	Alteração	Data
Igor Prata	Lançamento original	29/09/2017
Igor Prata	Alterações com bases em ensaios de voo	20/10/2017
Igor Prata	Alterações com bases em ensaios de voo	17/10/2017
Igor Prata	Revisões pós edição final da dissertação	27/10/2017

### 8.5.1 INSTALAÇÃO DO AMBIENTE SIMULADO.

Todas as ferramentas utilizadas nesse tutorial são baseadas nas soluções **DroneKit-Python** e no **Dronekit-sitl simulator** ferramentas *open source* mantidas pela comunidade de usuários do sistema Ardupilot.

### 8.5.2 INSTALAÇÃO

Antes instalar o PIP (instalador de pacotes python), certifique-se que os seguintes pacotes estão em seu S.O.:

```
$ sudo apt-get install python-matplotlib python-serial python-lxml
```

Para evitar incompatibilidades, retirar o pacote python-wxgtk2.8 do sistema.

Os sistemas **DroneKit-Python** e **Dronekit-sitl simulator** estão disponíveis para diversas plataformas Linux e podem ser instaladas por pacotes PIP. Se ainda não tiver, execute os comandos:

```
$ sudo apt-get install python-pip python-dev
```

Após atender os prerequisites, execute os comandos de instalação abaixo com os privilégios administrativos:

```
# pip install dronekit  
# pip install dronekit-sitl
```

Para mais instruções e verificar eventuais atualizações do Dronekit-Python e suas dependências, recomendamos que acesse a página como o passo-a-passo de instalação em:

**[http://python.dronekit.io/develop/sitl\\_setup.html](http://python.dronekit.io/develop/sitl_setup.html)**

### 8.5.3 CONFIGURAÇÃO INICIAL DO AMBIENTE SIMULADO

O DroneKit-SITL é a maneira mais simples e rápida para executar o Ardupilot em *Software in the Loop* (SITL) em ambientes Windows, Linux e Mac OS. A sua instalação é feita a partir de pacotes pela ferramenta Python PIP em todas as plataformas e funciona baixando binários pré-compilados de cada tipo de veículo, apropriados ao sistema operacional hospedeiro. O DroneKit-SITL ainda possui muitas funções experimentais e apenas alguns veículos são disponibilizados já compilados em suas bibliotecas.



```
vehicle = connect('127.0.0.1:14550', wait_ready=True)
```

E em seu GCS conecte pela segunda porta UDP configurada no MAVProxy: Na lista superior, certifique-se que o modo de conexão exibe a opção "**UDP**" em destaque, então selecione a opção "**Conexão**". Quando solicitado, digite o número da porta (nesse caso, a 14552).

Depois de conectado, o GCS exibirá os parâmetros do veículo simulado e a sua localização será exibida no mapa baseado nos parâmetros de carga do SITL.

O DroneKit-Python se comunica com o piloto automático do veículo usando o protocolo MAVLink, que por sua vez define como os comandos, telemetria e parâmetros e configurações são enviadas entre veículo, computador de bordo, estação em solo e outros sistemas, em uma rede MAVLink.

Algumas considerações gerais são necessárias com o uso do protocolo MAVLink:

- Mensagens e mensagens de confirmação (*message acknowledgments*) não possuem garantia de recebimento, o protocolo não é “*lossless*”.
- Comandos podem ser silenciosamente ignorados pelo piloto automático se esse não estiver em um estado que possa seguramente executá-las.
- Devido a natureza dessa rede, comandos de *acknowledgment* e *completion messages* não são enviados na maioria dos casos (e se enviados, podem não chegar).
- Comandos podem ser interrompidos antes da sua finalização completa.
- Diferentes pilotos automáticos podem interpretar comandos no protocolo de maneiras diferentes.
- Comandos podem chegar ao piloto automático por fontes diferentes.

#### 8.5.4 EXECUÇÃO DO AMBIENTE DE SIMULAÇÃO

A partir de um terminal *bash*, sem privilégios administrativos, é possível iniciar uma instância do DroneKit-SITL ao executar:

```
$ dronekit-sitl copter
```

Dessa maneira, o SITL será iniciado com as configurações básicas e oferecerá um veículo simulado que aceita conexões TCP pela porta 127.0.0.1:5760.

É possível especificar uma versão em particular do veículo, incluindo parâmetros como local de HOME, e modelo do veículo:

```
$ dronekit-sitl plane -3.3.0 --home=-35.363261,149.165230,584,353
```

Os seguintes exemplos, a depender da localização original da aeronave simulada que se deseja utilizar:

```
$ dronekit-sitl copter --home=lat , long , heading , alt
(urca):
$ dronekit-sitl copter --home=-22.9544,-43.1676,100,100
(forte Sao Joao):
$ dronekit-sitl copter --home=-22.945367,-43.156655,0,0
```

Recomenda-se, na primeira execução, carregar as configurações padrões e desabilitar o parâmetro **arming check**:

```
STABILIZE>param load ../Tools/autotest/copter_params.parm
STABILIZE>param set ARMING_CHECK 0
```

Note que isso deve ser usado apenas no começo para que se possa utilizar a ferramenta mais rapidamente. O SITL simula por padrão um veículo que não passa nos critérios de verificação dos parâmetros "*arming check*". Essa alteração faz com que o veículo seja aceito de maneira mais leniente, permitindo os exemplos a armarem e rodarem sem grandes esforços de configuração.

**Você nunca deve desabilitar o parâmetro "arming check" em um script ou veículo real!** Robotics (2016a)

Os argumentos básicos na execução do SITL são:

- `dronekit-sitl -h` (Lista todos os parâmetros to `dronekit-sitl`);
- `dronekit-sitl copter -h` (Lista os parâmetros adicionais para um veículo específico (nesse caso "copter"));
- `dronekit-sitl -list` (Lista todos os veículos disponíveis);
- `dronekit-sitl -reset` (Deleta todos os binários de veículos baixados);
- `dronekit-sitl ./path [args...]` (Inicia uma instância do SITL utilizando um arquivo específico de configuração).

### 8.5.5 COMPILANDO SITL A PARTIR DO SEU CÓDIGO FONTE

Opcionalmente pode-se nativamente compilar o SITL a partir de seu código fonte num ambiente Linux. Isso é útil quando deseja-se testar funcionalidades novas (ainda não liberadas nos pacotes de instalação estáveis). Também pode ser uma ferramenta importante caso alguns componentes do DroneKit-SITL na instalação normal não estejam funcionando corretamente.

O SITL instalado a partir do código-fonte possui algumas diferenças em relação ao DroneKit-SITL:

- O MAVProxy é incluído e iniciado por padrão;
- Pode-se utilizar o terminal MAVProxy controlar o piloto automático;
- Pode-se conectar o SITL via UDP pela porta 127.0.0.1:14550;
- Pode-se utilizar o comando de saída do MAVProxy para adicionar portas extras;
- Pode ser necessário desabilitar o parâmetro "**arming checks**" para executar os exemplos;
- É mais simples para adicionar *range finders* e *gimbals* virtuais.

#### 8.5.5.1 BAIXANDO O ARDUPILOT

Se você ainda não tem nenhuma cópia do repositório do Ardupilot na estação de trabalho que utilizará como ambiente de simulação, execute o comando:

```
$ git clone git://github.com/ArduPilot/ardupilot.git
$ cd ardupilot
$ git submodule update --init --recursive
```

#### 8.5.5.2 INSTALANDO PACOTES NECESSÁRIOS

Se estiver executando um sistema operacional baseado em Debian (como Ubuntu, ou Linux Mint) no *workstation* que será usado no ambiente de simulação, instale os seguintes pacotes:

```
$ sudo apt-get install python-matplotlib python-serial python-wxgtk3.0 python-wxtools python-lxml
```

```
$ sudo apt-get install python-scipy python-opencv ccache gawk
  git python-pip python-pexpect
$ sudo pip install future pymavlink MAVProxy
```

### 8.5.5.3 CONFIGURANDO O AMBIENTE

Adicione alguns caminhos no *path* de seu sistema operacional. insira as seguintes linhas ao final do arquivo ".bashrc" do seu diretório HOME (atenção ao "."no início do nome do arquivo).

```
export PATH=$PATH:$HOME/ardupilot/Tools/autotest
export PATH=/usr/lib/ccache:$PATH
```

Depois dessa alteração, recarregue o sistema de PATH utilizando o seguinte comando no terminal:

```
$. ~/.bashrc
```

### 8.5.5.4 INICIANDO O SITL SIMULATOR

Para iniciar o simulador, primeiramente altere o diretório para o do veículo que se deseja utilizar. Por exemplo, para executar como um veículo de asa rotativa, mude para o diretório:

```
$cd ardupilot/ArduCopter
```

Inicie o simulador com o comando "**sim\_vehicle.py**". A primeira vez que esse arquivo for executado, a opção "**-w**"deve ser inserida para limpar a memória EEPROM e carregar os parâmetros padrões do veículo.

```
$ sim_vehicle.py -w
```

Após a carga das configurações iniciais, o simulador deve ser executado normalmente. Primeiramente, termine o comando "sim\_vehicle.py"em execução com um Ctrl-C. Depois, digite no terminal:

```
$ sim_vehicle.py --console --map --aircraft test
```

#### **Licenciamento:**

Uso, reprodução e adaptações autorizados sob licença Creative Commons CC BY-SA

## ANEXO 6: PROCEDIMENTO DE INSTALAÇÃO E CONFIGURAÇÃO DO SISTEMA DE BORDO

O objetivo deste documento é prover uma série de passos necessários para a configuração do sistema operacional e dos serviços embarcados em *drones* e indispensáveis para o seu correto funcionamento.

Este manual leva em conta os projetos de *hardware* Raspberry Pi e Emlid Navio2 bem como diversos tutoriais e ferramentas referenciadas pela sua comunidade de usuários. Todos os programas descritos neste procedimento são *open source*.

O documento foi criado com base em VANTs do tipo quadrotor para sensoriamento de redes sem fio e aborda diferentes configurações focadas em projetos dessa natureza. Entretanto, boa parte do conhecimento é de propósito geral e poderá auxiliar pesquisadores que desejam utilizar *drones* de mesma natureza em suas pesquisas.

A concepção deste manual foi possível através da experiência coletiva de muitos alunos dos cursos de Mestrado e Doutorado do Instituto Militar de Engenharia (IME) com o intuito de auxiliar novos alunos no processo de montagem, configuração e organização de um VANT antes de um ensaio experimental em campo poupando tempo e mitigando riscos.

Antes de consultar esse manual, sugerimos fortemente que utilize também como referência, e se atente aos detalhes descritos (e provavelmente atualizados), o site do navegador NAVIO2 da empresa Emlid - utilizado nesse projeto. Toda a documentação do projeto está disponível no site do fabricante (EMLID, 2016)

Revisões		
Nome do Revisor	Alteração	Data
Igor Prata	Lançamento original	29/09/2017
Igor Prata	Alterações com bases em ensaios de voo	20/10/2017
Igor Prata	Alterações com bases em ensaios de voo	17/10/2017
Igor Prata	Revisões pós edição final da dissertação	27/10/2017

### 8.6.1 INSTALANDO O SISTEMA OPERACIONAL

Baixar a imagem de disco mais recente do Sistema Operacional Raspbian, especialmente customizada pela Emlid para a placa NAVIO2 em:

<https://docs.emlid.com/navio2/common/ardupilot/configuring-raspberry-pi/>

Descompacte o arquivo baixado. Por exemplo, pelo comando:

```
$ unxz emlid-raspbian-20170323.img.xz
```

Checar o conteúdo baixado com o MD5 Sum do arquivo em:

```
$ wget https://files.emlid.com/images/MD5SUMS
```

Por exemplo, com o comando (dependendo do arquivo baixado e do checksum do arquivo):

```
$ md5sum emlid-raspbian-20170323.img | grep  
efd367aa19eb217647816f636cabb6
```

Grave a imagem (baixada, descompactada e verificada) em um cartão microSD de no mínimo 16GB de tamanho e boa velocidade de acesso (classe 10). Utilize o cartão em seu computador com o auxílio de um adaptador (USB ou SD):

```
$ dd bs=4M if=emlid-raspbian-20170323.img of=/dev/sdX conv=fsync
```

Alterar o nome da imagem para o arquivo baixado e o destino (of) para o caminho do cartão SD correto. Se preferir uma ferramenta de modo gráfico para a gravação da imagem de disco, sugerimos o programa ETCHER (VIOTTI, 2016), dessa maneira não será necessário descompactar o arquivo de imagem:

*Sugerimos criar partições separadas para salvar dados que podem consumir muito espaço com o tempo. Arquivos como logs, banco de dados e vídeos devem ser salvos em partições específicas (idealmente em unidades de armazenamento separadas, como pendrives) para que não lotem a partição do sistema operacional em voo e prejudiquem a missão.*

### 8.6.2 CONFIGURANDO O ACESSO WIFI

O Raspberry pi3 possui um módulo interno de WiFi mas diversos *dongles* WiFi USB são suportados nativamente. Uma grande lista de hardwares suportados pode ser acessada aqui. Recomendamos fortemente que embarque em seu VANT uma adaptador de rede

WiFi externo que suporte 5GHz e não conte apenas com o módulo interno. Este adaptador poderá ser utilizado em conjunto para sensoriamento ou para *broadcasting* de dados.

Se quiser usar o sistema operacional embarcado sem necessitar de cabos HDMI, ethernet ou um teclado, configure a rede WiFi do roteador que será utilizado nos ensaios, editando o arquivo **wpa\_supplicant.conf** localizado no cartão SD card (na partição **/boot**). Para adicionar a sua rede, basta editar as seguintes linhas deste arquivo antes de colocar o cartão SD no *Drone*.

```
network={
  ssid="yourssid "
  psk="yourpasskey "
  key_mgmt=WPA-PSK
}
```

*Lembre-se de salvar o arquivo pelo editor antes de fechar.*

### 8.6.3 TESTANDO AS CONFIGURAÇÕES DE REDE

Após essa configuração, desmonte a unidade (**/boot**), retire o adaptador de seu computador pessoal, retire o cartão microSD do adaptador e insira o mesmo no *slot* microSD na placa Raspberry pi3 com o auxílio de uma pinça.

Energize o computador de bordo com um módulo externo. ou *Power Bank*, pela porta MicroUSB B da Raspberry pi3.

Para localizar o novo endereço IP obtido pelo computador de bordo, pode-se utilizar a interface de gerência de seu roteador WiFi ou utilizar a ferramenta NMAP, pelo comando:

```
$ nmap -sn 192.168.1.*
```

Procure pelo *hostname* "NAVIO".

Se preferir uma interface gráfica, pode utilizar a aplicação Zenmap (Windows ou Linux) ou Fing (em seu telefone).

Uma vez com o IP identificado, tente se conectar por SSH pelo comando:

```
$ ssh pi@navio.local
```

Troque o nome **navio.local** pelo IP obtido anteriormente.

A partir de computadores Windows, pode utilizar a ferramenta Putty para se conectar por SSH.

#### 8.6.4 PROBLEMAS CONHECIDOS

Se mesmo estando seguro que o Raspberry pi3 foi corretamente configurado para acessar a rede (e estiver obtendo um endereço IP próprio), mas ao tentar conectar via SSH, receber o erro abaixo:

```
SSH2_MSG_KEXINIT sent
Connection closed by 192.168.1.*
```

O problema talvez esteja nas chaves padrões do Serviço SSH no Raspberrian. Para tal será necessário zerar essas chaves no OpenSSH Server no computador de bordo. Faça o seguinte comando no Raspberry Pi:

```
$ sudo rm /etc/ssh/ssh_host_* -v
$ sudo dpkg-reconfigure openssh-server
```

E tente conectar novamente por SSH a partir de sua estação de trabalho.

#### 8.6.5 CONFIGURANDO O S.O. E OS PRINCIPAIS SERVIÇOS

Mais detalhes sobre os procedimentos abaixo podem ser encontrados em:

<https://www.raspberrypi.org/documentation/configuration/raspi-config.md>

Durante o primeiro login no Raspbian, execute o comando para configurações iniciais do sistema:

```
$ sudo raspi-config
```

O menu da figura 8.27 será exibido:

Nele, as seguintes opções devem ser selecionadas e executadas:

- a) **Expand Filesystem:** Use essa opção para garantir que todo o espaço disponível no cartão SD será utilizado pelo S.O.;
- b) **Change User Password:** Mude a senha do usuário padrão do sistema (pi);
- c) **Internationalisation Options:** Configure o idioma e as opções regionais - Para Selecionar o timezone correto no Brasil, escolher Sao\_Paulo;
- d) **Enable Camera:** Habilite a câmera do Raspberry Pi nessa opção;
- e) **Advanced Options:** Entre em *advanced settings* e habilite o serviço SSH.

Ao final, sair do menu e, de volta ao terminal, reiniciar o computador:

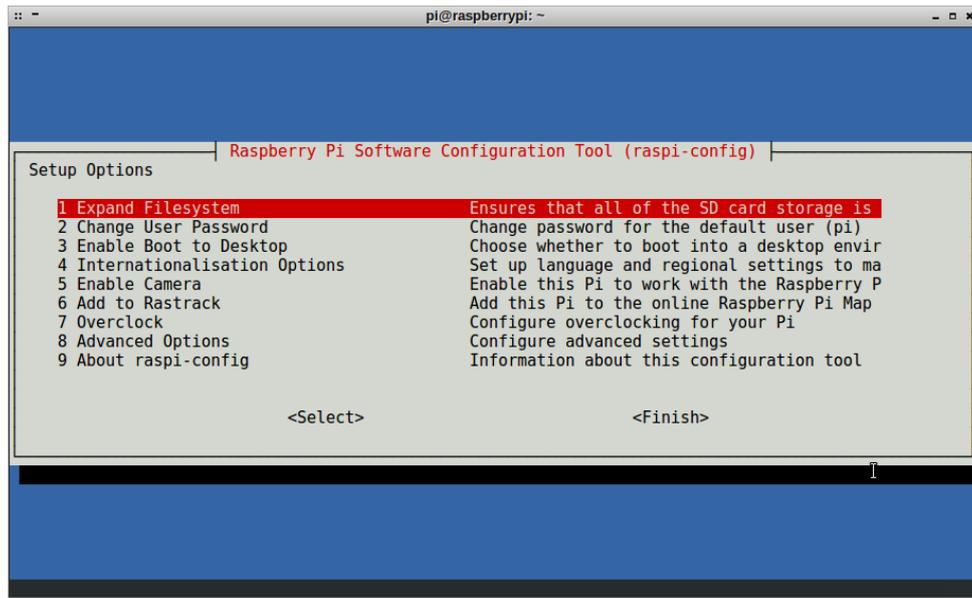


FIG. 8.27: Tela inicial do comando Rasp-Config

```
$ sudo reboot
```

Depois de reiniciar e logar com a sua nova senha, atualize todo o sistema operacional pela internet com o comando:

```
$ sudo apt-get update && sudo apt-get dist-upgrade
```

Configure os pontos de montagem de suas partições extras (se houver), no arquivo `/etc/fstab`.

### 8.6.6 CONFIGURANDO OS SERVIÇOS ESSENCIAIS

Uma versão mais detalhada desse processo pode ser consultada em:

<https://docs.emlid.com/navio2/common/ardupilot/installation-and-running/>

Para configurar o serviço ArduPilot, indispensável para o *drone*, execute o comando abaixo para definir o modelo de aeronave desejada:

```
$ sudo update-alternatives --config arducopter
```

E escolha a opção:

```
15 /opt/ardupilot/navio2/arducopter-3.4/bin/arducopter-quad
```

Configure o computador de solo pelo comando:

```
$ sudo nano /etc/default/arducopter
```

E habilite todos os *endpoints* necessários para telemetria:

```
TELEM1="--A_udp:192.168.1.6:14550 "  
TELEM2="--C_/dev/ttyAMA0"
```

Troque o endereço IP pelo de seu computador que servirá como estação de solo. Se houver um *tablet* ou celular que será utilizado como estação de solo extra, configure como TELEM3 e habilite na linha.

```
$ARDUPILOT_OPTS="$TELEM1_$TELEM2"
```

Salve o arquivo e feche o editor.

Para habilitar o serviço Arducopter a iniciar automaticamente em todos os *boots*, execute o comando:

```
$ sudo systemctl enable arducopter
```

Opcionalmente, pode-se executar o comando abaixo para uma interface consolidada de configurações do ArduPilot:

```
$ sudo emlidtool ardupilot
```

Se você alterar alguma configuração posteriormente, execute o comando abaixo para que ela tenha efeito:

```
$ sudo systemctl daemon-reload
```

Para iniciar o serviço ArduPilot:

```
$ sudo systemctl start arducopter
```

Para interromper o serviço Ardupilot em execução:

```
$ sudo systemctl stop arducopter :
```

Para desabilitar o carregamento automático:

```
$ sudo systemctl disable arducopter
```

Para verificar o estado do serviço:

```
$ systemctl is-enabled arducopter
```

É necessário adicionar o usuário principal ao grupo "dialout" do sistema:

```
$ sudo adduser pi dialout
```

### 8.6.6.1 *SETUP* MANDATÓRIO DO ARDUPILOT

Caso receba erro que impeçam a ativação dos motores e não dar mais erros na conexão entre o GCS e o ArduCopter, na interface do APM Planner (por exemplo), clique no botão "**config/tuning**"

- a) Selecione a aba "**full parameter list**";
- b) Altere o parâmetro **ARMING\_CHECK = 0**;
- c) Depois selecione a opção: **write to**.

### 8.6.6.2 CONFIGURAÇÃO DE ENXAMES (*SWARM*)

Se houver múltiplas aeronaves para a missão ou caso o GCS tenha que controlar aeronaves diferentes:

- a) Selecione a aba "**full parameter list**";
- b) Modificar parâmetro **SYSID\_THISMAV = 0**, atribuindo um valor diferente para cada veículo;
- c) Depois selecione a opção: **write to**.

*Numere em acordo com os demais alunos para facilitar integração com os usuários do mesmo canal de rádio.*

### 8.6.6.3 CONFIGURAÇÃO DO SONAR MAXBOTIX I2C

A solução proposta de VANT utiliza um sonar interligado na placa Navio2 pela porta I2C. Para o seu correto funcionamento deve-se editar o arquivo:

```
$ sudo nano /boot/config.txt
```

Buscar a linha contendo o parâmetro:

```
dtparam=i2c1_baudrate=1000000
```

Alterar o valor de 1M para 400K:

```
dtparam=i2c1_baudrate=400000
```

Na janela de "Initial Setup" do APM Planner2 (por exemplo), na parte de "**Optional Setup**", "**Range Finder**", escolher a opção **APM2-MaxbotixI2C**, como mostrado na figura 8.28:

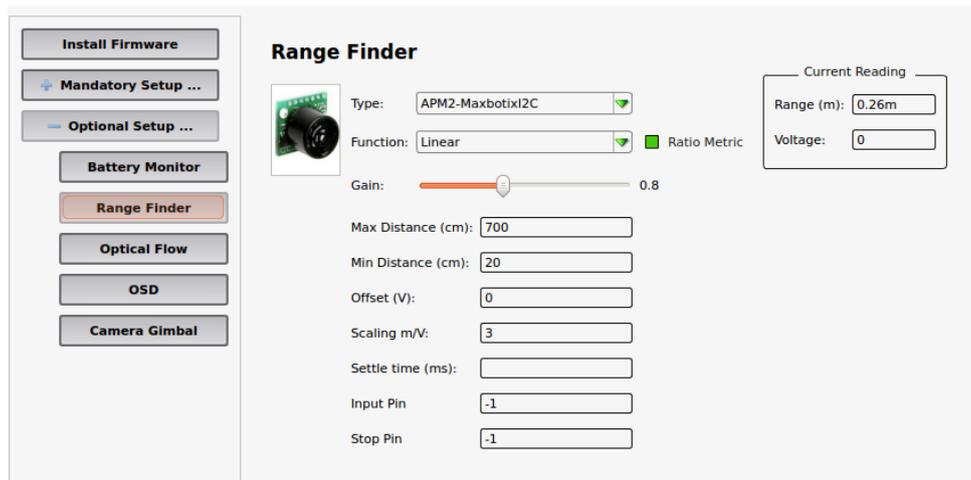


FIG. 8.28: Tela de configuração de *Range Finder* no GCS APM Planner2

#### 8.6.6.4 CONFIGURAÇÃO DO RANGE FINDER (GARMIN LIDAR LITE V3)

Caso esteja usando um *range finder* do tipo LIDAR (*Light Detection And Ranging*) pela porta I2C no lugar de um sonar, mais especificamente, se for o modelo Garmin Lidar Lite V3, execute os passos abaixo para configuração correta do dispositivo:

Para versões do ArduCopter inferiores à 3.4, no seu GCS clique no botão "**config/-tuning**":

- a) Selecione a aba "**full parameter list**";
- b)  $RNGFND\_TYPE = 3$  (I2C Driver, para Lidar V2 e versões iniciais do V3)
- c)  $RNGFND\_TYPE = 15$  (I2C Driver, para o Lidar V3 no ArduCopter 3.5-rc2 e superiores)
- d)  $RNGFND\_MIN\_CM = 20$  (20cm)
- e)  $RNGFND\_MAX\_CM = 4000$  (4000cm = 40m)
- f)  $RNGFND\_OFFSET = 0$
- g)  $RNGFND\_GNDCLEAR = 10$  (A distância do Lidar até o chão quando a aeronave está pousada. Mínimo recomendado 20cm);
- h)  $RNGFND\_SCALING = 1$  (ajuste entre 0.8 - 1.0)
- i)  $RNGFND\_GAIN = 0.5$  (Aumente esse valor para fazer o veículo responder mais rapidamente a a mudanças de altitude em relação ao solo. Porém, valores altos podem fazer com que o controle fique instável);

- j) `EK2_ALT_SOURCE = 1` (Utilize o sensor de *RangeFinder* para medidas de altitude. Essa opção funciona nos modos de voo **Alt\_Hold**, **PosHold** e **Loiter**. Se o valor for 0 utilizará o barômetro e 2, o GPS);
- k) Depois selecione a opção: **write to**.

Para versões do ArduCopter 3.4 ou superiores, no seu GCS clique no botão "**config/tuning**":

- a) Selecione a aba "**full parameter list**"
- b) `EK2_RNG_USE_HGT = 50` (Percentual de 0 a 100% do parâmetro `RNGFND_MAX_CM`. O *Rangefinder* se tornará a fonte de medição de altitude primária se a aeronave estiver abaixo desse nível. Por exemplo: se `RNGFND_MAX_CM = 4000` e o `EK2_RNG_USE_HGT = 50`, então o *RangeFinder* se tornará o sensor principal abaixo de 20m de altitude. APENAS USE ISSO SE ESTIVER COM VOOS LENTOS programados. Por padrão o valor é -1 o que desabilita o *RangeFinder* como fonte de medição);
- c) `EK2_ALT_M_NSE = 3` (Ruído do Barômetro em metros. Aumente esse parâmetro para depender menos desse sensor e mais nos acelerômetros e GPS)
- d) `EK2_RNG_M_NSE = 0.5` (Aumente esse parâmetro se houver muito ruído no sensor de *rangefinder*)
- e) `EK2_TERR_GRAD = 0.1` (Aumente esse parâmetro se o chão abaixo do veículo não for plano)
- f) `EK2_RNG_I_GATE = 500` (Se as medidas do *rangefinder* estiverem fora dessa quantidade, o filtro EKF rejeita os dados de distância. Deixe os valores *default* a não ser que saiba o que está fazendo!)
- g) `EK2_HGT_I_GATE = 500` (Veja acima)
- h) Depois selecione a opção: **write to**.

*Não altere o parâmetro original `dtparam=i2cl_baudrate=1000000` em `/boot/-config.txt` caso esteja usando um LIDAR!*

### 8.6.7 CONFIGURAÇÃO DE GIMBAL

Se tiver uma estrutura robótica de Gimbal, alimente a placa do Navio2 com uma linha extra de 5 volts através de um UBeck ligado à placa de distribuição de corrente do *drone*. Faça isso antes de ligar os motores de passo as portas da Navio2. A configuração no APM Planner segue como a figura 8.29.

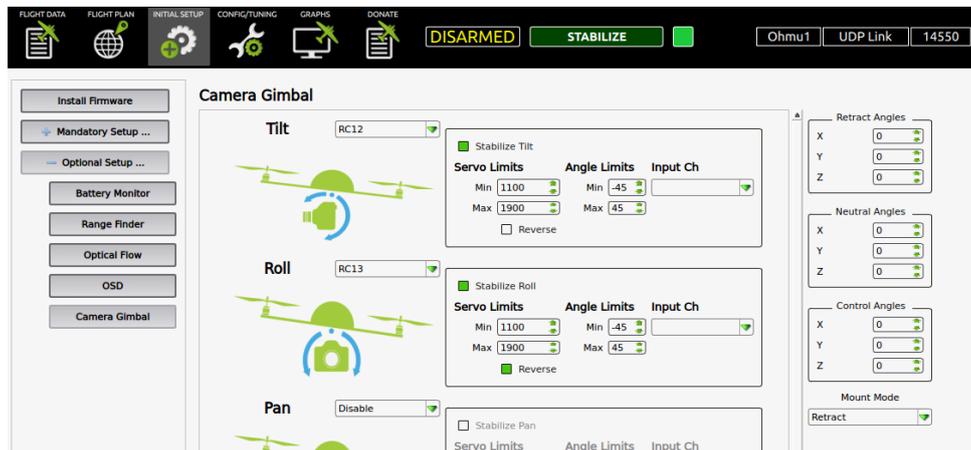


FIG. 8.29: Tela de configuração de Gimbal no GCS APM Planner2

### 8.6.8 REVISAR CONFIGURAÇÕES DE ROTORES

Com o Controle RC armado, verificar com o tato ou com a ajuda de uma folha de papel o sentido de rotação de cada um dos rotores. Para a sua segurança, mantenha o nível de *thrust* baixo! Os sentidos corretos de cada motore pode ser verificado na figura 8.30

### 8.6.9 DEPENDÊNCIAS DE SOFTWARE

As seguintes bibliotecas são necessárias para a execução dos programas a bordo da aeronave:

- **Scipy:**

```
$ sudo apt-get install python-scipy
```

- **Numpy:**

```
$ sudo pip install numpy
```

- **Módulo WiFi (IWCONFIG) para Python:**

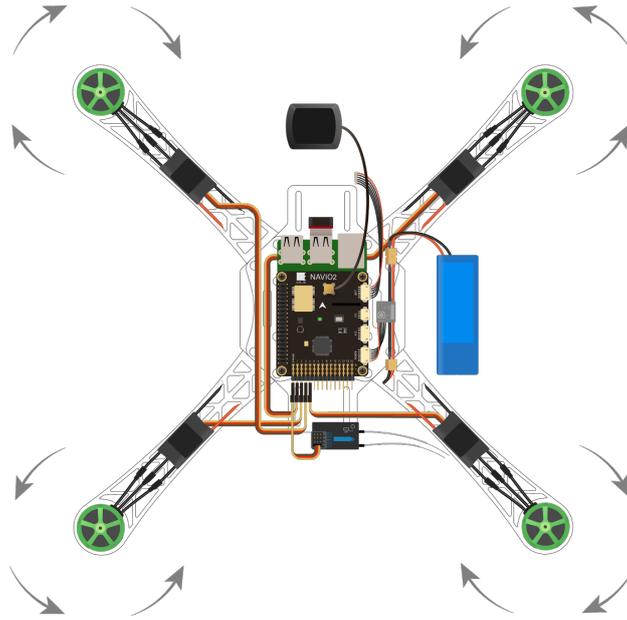


FIG. 8.30: Sentido de rotação dos rotores (Fonte: Emlid)

```
$ sudo pip install WiFi
```

- **BT-Proximity para Python:**

```
$ git clone https://github.com/ewenchou/bluetooth-proximity.git
$ cd bluetooth-proximity
$ sudo python setup.py install
```

#### 8.6.10 FERRAMENTAS BLUETOOTH

A customização do Sistema Operacional Raspbian fornecida pela Emlid desabilita o módulo bluetooth *onboard*, obrigado a utilização de *dongles* USB para lançar mão desse hardware. A maior parte dos adaptadores bluetooth não precisam de instalações de módulos especiais para funcionarem em Linux, mas algumas ferramentas deverão ser instaladas caso queira utilizar os seus recursos.

Instale as seguintes dependências pelo comando:

```
$ sudo apt-get install bluemon bluez bluez-tools bluetooth bluez-hcidump bluez-test-scripts libbluetooth-dev libudev-dev
```

```
python-bluez hcidump
```

Diretório com *scripts* BLUEZ de exemplo, após a instalação, se encontra em:  
`/usr/share/doc/bluez-tests/examples`

### 8.6.11 FERRAMENTAS WIFI

O módulo *onboard* do Raspberry pi3 funciona normalmente. mesmo com as customizações de *kernel* providas pela Emlid, o que permite utilizar ele em parceria com outras placas de rede (para diferentes propósitos) pelas portas USB. Porém será necessário instalar as bibliotecas WiFi do Python se quiser utilizar os *scripts* de coleta:

```
$ sudo pip install WiFi
```

### 8.6.12 FERRAMENTAS ZWAVE

ZWave é um protocolo comercial para automação residencial e IoT. Se possuir um dongle compatível ZWave que seja reconhecido no Linux, precisará das seguintes ferramentas para desenvolver ferramentas para ele:

```
$ sudo apt-get install --force-yes -y make libudev-dev g++  
  libyaml-dev  
$ pip install python_openzwave  
$ pip install urwid
```

E configurar a interface serial associada ao rádio com o comando:

```
$ pyozw_check -i -d /dev/ttyUSB0
```

Se o dispositivo `/dev/ttyUSB0` for como o sistema reconheceu o seu dispositivo.

### 8.6.13 CONFIGURAR PLACAS DE REDES

Os procedimentos abaixo são orientados para os projetos de sensoriamento de redes sem fio e não são mandatórios para a configuração do *drone*, prossiga apenas se for necessário instalar as seguintes placas de rede.

### 8.6.14 TP-LINK T2UH (RALINK MT7610U)

A maior parte dos *dongles* USB possuem módulo pré configurados e carregados em S.O.s Linux. Porém alguns hardwares WiFi no padrão IEEE 802.11 ac precisarão de novos

módulos compilados.

Para configurar o dongle TP-LINK T2UH Wireless adapter Driver (Ralink mt7610u) No Raspberry pi3 execute os comandos abaixo para baixar os arquivos necessários:

```
$ sudo apt-get install git build-essential
$ git clone https://github.com/Myria-de/
  mt7610u_WiFi_sta_v3002_dpo_20130916.git
```

Para compilar e instalar o novo módulo de rede, siga os passos abaixo:

```
$ cd mt7610u_WiFi_sta_v3002_dpo_20130916
$ make
$ sudo make install
$ sudo mkdir -p /etc/Wireless/RT2870STA
$ sudo cp RT2870STA.dat /etc/Wireless/RT2870STA/RT2870STA.dat
```

*Reinicie o S.O. do drone.*

Execute o comando **ifconfig** e verifique se a nova interface de rede está disponível.

#### 8.6.15 TP-LINK T4UH (AC1200)

A maior parte dos *dongles* USB possuem módulo pré configurados e carregados em S.O.s Linux. Porém alguns hardwares WiFi no padrão IEEE 802.11 ac precisarão de novos módulos compilados.

A instalação do adaptador de rede TP-Link T4UH (AC1200) deve seguir como descrito abaixo, dependendo do Sistema Operacional onde deseja utilizá-la. **No Ubuntu 16.04:** Baixar e instalar o pacote .DEB em:

<https://www.ubuntuupdates.org/package/core/xenial/universe/base/rtl8812au-dkms>  
Instale com o comando:

```
$ sudo dpkg -i rtl8812au-dkms_4.3.8.12175.20140902+dfsg-0
  ubuntu2_all.deb
```

**No RaspBerry Pi:** Baixar os drivers e seguir os passos descritos em:

<https://github.com/ptpt52/rtl8812au>

Instale os *kernel headers* e outras dependências.

```
$ sudo apt-get install linux-image-rpi-rpfbv linux-headers-rpi-
  rpfbv dkms build-essential bc
```

Adicione as configurações ao final do arquivo `/boot/config.txt`, e reinicie o Raspberry Pi:

```
kernel=vmlinuz-3.10-3-rpi
initramfs initrd.img-3.10-3-rpi follow kernel
```

Edite o arquivo **Makefile** e configure os seguintes parâmetros:

```
CONFIG_PLATFORM_I386_PC = n
CONFIG_PLATFORM_ARM_RPI = y
```

Execute os seguintes comandos:

```
$ cd /usr/src/rtl8812au
$ sudo make clean
$ sudo make
$ sudo make install
$ sudo modprobe -a rtl8812au
```

## 8.6.16 COMANDOS ÚTEIS

Abaixo são apresentados alguns comandos de terminal úteis para a verificação de estados dos rádios e que auxiliarão em análises, configurações, pareamento e habilitação de recursos ao operador.

## 8.6.17 COMANDOS BLUETOOTH

Não deixar de instalar as seguintes dependências no Raspberrian:

```
$ sudo apt-get install bluetooth bluez bluez-tools rfkill rfcomm
```

Para garantir que o serviço Bluetooth no Linux está rodando, execute:

```
$ sudo service bluetooth start
```

Para verificar se alguma interface de rede está bloqueada por alguma operação:

```
$ sudo rfkill list
$ sudo rfkill unblock bluetooth
```

Se alguma operação não estiver funcionando com a interface bluetooth, pode-se desabilitar e habilitar a interface para verificar se o comportamento se normaliza:

```
$ sudo hciconfig hci0 down
$ sudo hciconfig hci0 up
```

Se a interface que estiver verificando for a primeira (hci0).

Para verifica as interfaces, endereço MAC e o estado, execute:

```
$ hciconfig
```

Mais sucintamente, para descobrir o endereço MAC dos dispositivos bluetooth locais:

```
$ hcitool dev
```

Para verificar os dispositivos bluetooth pareáveis ao redor, execute:

```
$ hcitool scan
```

Algumas maneiras rápidas de capturar a intensidade de sinal RSSI dos dispositivos bluetooth pareados é:

```
# btmon |grep RSSI & sudo hcitool scan
# hcitool rssi <Target device MAC address>
# watch -n 1 hcitool rssi <Target device MAC address>
```

Verifica o estado de um dispositivo bluetooth específico nas proximidades:

```
$ sudo l2ping <Target device MAC address>
```

Em um terminal Linux, existem algumas formas de parear e se conectar a um dispositivo bluetooth (mandatório para captura de sinais RSSI) nas proximidades que esteja apto a isso:

```
# hcitool cc <Target device MAC address>
# gatttool -b <Target device MAC address> -t random --
  interactive
```

Se isso não funcionar, pare e remova o dispositivo local, e então tente novamente. Por exemplo:

```
# sudo hciconfig hci0 down
# hciconfig hci0 up
# gatttool -b <Target device MAC address> -t random --
  interactive
```

Outra forma de se conectar:

```
# rfcomm connect 0 <Target device MAC address>
```

Por exemplo:

```
$ sudo rfcomm connect hci0 <Target device MAC address> 2 &
```

Para verificar os dispositivos já pareados e outros estados do sistema de gerenciamento bluetooth local:

```
# bluetoothctl
```

Depois, para conectar:

```
connect <Target device MAC address>
```

### 8.6.18 COMANDOS BLUETOOTH BLE

Para verificar os dispositivos Bluetooth BLE (Bluetooth Low Energy) ao redor, execute o comando:

```
$ sudo hcitool lescan
```

Para parear com um dispositivo BLE (não é mandatório para capturar sinal de RSSI), execute o comando:

```
$ sudo hcitool lewladd <Target device MAC address>
```

Talvez seja necessário para alguns dispositivos BLE:

```
$ sudo hcitool lecc <Target device MAC address>
```

```
$ gatttool -t random -b <Target device MAC address> -I
```

Se o resultado for como o seguinte, a conexão foi bem-sucedida:

```
[<Target device MAC address>][LE]> connect Attempting to connect  
to EA:FB:B5:CE:B0:13 Connection successful [EA:FB:B5:CE:B0  
:13][LE]> characteristics handle: 0x0002, char properties: 0  
x0a, char value handle: 0x0003, uuid: 00002a00  
-0000-1000-8000-00805f9b34fb
```

Em caso de erro, pode tentar habilitar e desabilitar os dispositivos bluetooth locais, como explicado anteriormente.

Algumas maneiras rápidas de capturar a intensidade de sinais (RSSI) de dispositivos BLE são:

```
$ sudo btmon & hcitool lescan
$ sudo btmon & hcitool lescan --duplicates
```

Captura os dados bluetooth brutos (RAW) da interface:

```
$ hcidump -R
```

Para não reconectar a um dispositivo Bluetooth a toda nova inicialização do sistema da aeronave, pode-se registrar o dispositivo pareado no arquivo:

```
$ sudo nano /etc/systemd/system/dbus-org.bluez.service
```

### 8.6.19 COMANDOS WIFI

Uma breve lista de comandos WiFi para terminal em Linux que poderão auxiliar na configuração, verificação de problemas e automatização, pode ser vista a seguir:

Para verificar quais são as redes sem fio no padrão IEEE 802.11 disponíveis para a interface de rede de nome "wlan0":

```
$ sudo iwlist wlan0 scan
```

Antes, execute o comando **ifconfig** para saber qual interface de rede deseja utilizar em sua verificação.

- **iwconfig** (Manipula os parâmetros WiFi básicos);
- **iwlist** (Permite iniciar uma varredura por redes disponíveis, listar as frequências, bit-rates, chaves de criptografia e etc.);
- **iwspy** (Permite detectar a qualidade por link node);
- **iwpriv** (Permite manipular as extensões *wireless* específicas de um *driver*);
- **ifrename** (Permite renomear interfaces baseado em critérios).

Algumas maneiras rápidas de capturar a intensidade de sinais (RSSI) de dispositivos WiFi são:

```
$ cat /proc/net/wireless
$ watch -n 1 cat /proc/net/wireless
```

Se precisar de uma ferramenta de terminal Linux que disponibilize uma interface gráfica que monitore a placa de rede, pode-se baixar e instalar o programa "wavemon"

```
$ sudo apt-get install wavemon
$ sudo wavemon
```

Para checar as configurações de interfaces de redes no Raspberry Pi você pode listar o conteúdo do arquivo de configuração com o comando:

```
$ cat /etc/network/interfaces
```

Para se conectar a uma rede WiFi específica pelo terminal:

```
# wpa_passphrase <Your_SSID> <Your_passwd> > wpa_supplicant.conf
# wpa_supplicant -B -i <wlan0> -c wpa_supplicant.conf -D wext
# wpa_supplicant -B -i <wlan0> -c wpa_supplicant.conf -D nl80211
# dhclient <wlan0>
```

### **Licenciamento:**

Uso, reprodução e adaptações autorizados sob licença Creative Commons CC BY-SA