

UNIVERSIDADE FEDERAL DE ITAJUBÁ

PROGRAMA DE PÓS-GRADUAÇÃO EM
CIÊNCIA E TECNOLOGIA DA COMPUTAÇÃO

Integração de *Software* e *Hardware* no Desenvolvimento de
VANTs Autônomos de Baixo Custo

Luiz Flávio Felizardo

Itajubá, Janeiro de 2016

UNIVERSIDADE FEDERAL DE ITAJUBÁ

PROGRAMA DE PÓS-GRADUAÇÃO EM
CIÊNCIA E TECNOLOGIA DA COMPUTAÇÃO

Luiz Flávio Felizardo

Integração de *Software* e *Hardware* no Desenvolvimento de
VANTs Autônomos de Baixo Custo

Dissertação submetida ao Programa de Pós-graduação em
Ciência e Tecnologia da Computação como parte dos requisitos
para obtenção do Título de Mestre em Ciência e Tecnologia da
Computação

Área de Concentração: Matemática Computacional

Orientador: Prof. Dr. Alexandre Carlos Brandão Ramos

Coorientador: Prof. Dr. Marcos Theiss Neves

Janeiro de 2016
Itajubá – MG

Agradecimentos

A toda equipe, que cooperou de alguma forma com este trabalho, incluindo os colegas e colaboradores da *Black Bee Drones*, sem os quais este caminho teria sido muito mais lancinante.

Aos professores Dr. Alexandre Carlos Brandão Ramos e Dr. Marcos Theiss Neves, pela dedicação, orientação e paciência neste período de aprendizado.

Aos meus colegas do mestrado, em especial ao Armando Ferraz Graça Neto, que me ajudaram neste tempo de estudo e pesquisa.

A todos que me apoiaram, pelo tempo que deixamos de estar juntos...

À minha mãe, Luiza; a ela todos os créditos...

Dedico

Resumo

INTEGRAÇÃO DE *SOFTWARE* E *HARDWARE* NO DESENVOLVIMENTO DE VANTs AUTÔNOMOS DE BAIXO CUSTO

Hoje em dia, Veículos Aéreos Não Tripulados (VANTs) são amplamente utilizados em um grande número de aplicações desde tarefas militares, como a coleta de informação, até aplicações civis, tais como: monitoramento, inspeção e até mesmo resgate. A principal contribuição deste trabalho de pesquisa é a substituição de métodos tradicionais de inspeção e filmagem aérea por métodos que utilizem VANTs. Vantagens, como o baixo custo de operação e, principalmente, o baixo risco de acidentes quando comparados aos métodos tradicionais, viabilizaram este trabalho. As aeronaves utilizadas foram capazes de fazer procedimentos de pouso e decolagem autônomos, mantendo a rota previamente programada, filmando e enviando imagens ao vivo para a estação de controle em terra, assim como possibilitaram a gravação de todos os dados para posterior processamento ou análise. Além disso, como contribuição adicional, nesta pesquisa, foram detalhados os aeromodelos escolhidos, o sistema utilizado, suas configurações e também algumas aplicações desenvolvidas.

Abstract

SOFTWARE AND HARDWARE INTEGRATION IN THE DEVELOPMENT OF LOW COST AUTONOMOUS UAVS

Nowadays Unmanned Aerial Vehicles (UAVs) are widely used in a vast number of applications from military tasks like gathering information to civil applications such as monitoring, research and rescue operations. This research proposes the replacement of traditional aerial inspection and filming methods by one that uses autonomous Unmanned Aerial Vehicles (UAVs). Advantages like low cost of operation and particularly the low risk of accidents compared to traditional methods make this work feasible. The aircrafts used were capable of executing autonomous landing and takeoff procedures also maintaining the previously programmed route and sending live images to the control station on the ground. All the flight data was stored for further processing or analysis. Furthermore, some additional contribution were tackled on this research like better choice of the aircraft model and configuration, the application developed and the computational system settings used.

Sumário

AGRADECIMENTOS.....	III
RESUMO	IV
ABSTRACT	V
SUMÁRIO.....	VI
LISTA DE FIGURAS	VII
LISTA DE TABELAS	VIII
SIGLAS	IX
1 INTRODUÇÃO.....	10
1.1 CLASSIFICAÇÃO	11
1.2 REGULAMENTAÇÃO NO BRASIL.....	13
1.3 OBJETIVOS.....	14
1.4 ORGANIZAÇÃO DO TRABALHO.....	15
2 FUNDAMENTAÇÃO TEÓRICA	16
2.1 SISTEMAS AÉREOS NÃO TRIPULADOS	17
2.1.1 ARDUPILOT.....	17
2.1.2 PAPANAZZI	19
2.1.3 OUTROS SISTEMAS.....	19
2.2 O <i>HARDWARE</i> DO SISTEMA COMPUTADORIZADO ESCOLHIDO	20
2.2.1 AUTOPILOTO E MICROCONTROLADORES.....	20
2.2.2 UNIDADE DE MEDIDA INERCIAL (IMU).....	22
2.2.3 SISTEMA DE POSICIONAMENTO GLOBAL (GPS).....	23
2.2.4 <i>MODEMS</i> DE COMUNICAÇÃO	24
2.2.5 RÁDIO, RECEPTOR RC E CONVERSOR DE SINAIS.....	25
2.3 O <i>SOFTWARE</i> DO SISTEMA COMPUTADORIZADO ESCOLHIDO.....	28
2.3.1 CENTRAL PAPANAZZI.....	28
2.3.2 ESTAÇÃO DE CONTROLE EM TERRA (GCS).....	29
2.4 INTELIGÊNCIA ARTIFICIAL E REDES NEURAS ARTIFICIAIS	32
3 INTEGRAÇÃO DE SOFTWARE E HARDWARE.....	34
3.1 AERONAVES.....	35
3.1.1 ASA FIXA.....	35
3.1.2 ASA ROTATIVA	36
3.2 MONTAGEM E CONFIGURAÇÃO DO <i>HARDWARE</i> E DO <i>SOFTWARE</i>	40
3.3 VÍDEO E GRAVAÇÃO DE IMAGENS EM VANTS	43
3.4 CONFIGURAÇÃO DE VÍDEO UTILIZADA	45
3.4.1 <i>HARDWARE</i> DO SISTEMA DE VÍDEO	45
3.4.2 <i>SOFTWARE</i> DO SISTEMA DE VÍDEO	48
4 APLICAÇÕES DE VANTS AUTÔNOMOS DE BAIXO CUSTO	50
4.1 REDE NEURAL ARTIFICIAL E ANÁLISE DE IMAGEM TÉRMICA	51
4.2 MAPAS KOHONEN E PROCESSAMENTO DE IMAGENS AÉREAS	55
5 CONCLUSÃO.....	60
REFERÊNCIAS.....	61

Lista de Figuras

Figura 1 - Exemplo de inspeção usando o método tradicional	14
Figura 2 - Uma Placa Ardupilot 2.5 Disponível no Mercado	18
Figura 3 - Um Exemplo de Interface do Sistema Ardupilot	18
Figura 4 - Lisa/M V2.0, visão por cima com pinagem.....	21
Figura 5 - Lisa/M V2.0, visão por baixo.....	21
Figura 6 - Aspirin IMU, visão de cima.....	22
Figura 7 - Aspirin IMU, visão de baixo.....	23
Figura 8 - GPS U-blox LEA 5S	24
Figura 9 - Modem XBee Pro ZB com saída pigtail	24
Figura 10 - Modem XBee Pro ZB com antena pequena	25
Figura 11 - Rádio com chave de três posições.....	26
Figura 12 - Receptor do rádio	26
Figura 13 - Placa ATmega PPM <i>Encoder</i>	27
Figura 14 - Paparazzi Center.....	28
Figura 15 - Trajetória oval mostrada na GCS.....	31
Figura 16 - Primeiro aeromodelo de asa fixa (monomotor)	35
Figura 17 - Segundo aeromodelo de asa fixa (bimotor)	36
Figura 18 - Ação das correntes de ar gerada pelo helicóptero	37
Figura 19 - Helicóptero utilizado (Classe 450).....	38
Figura 20 - Helicóptero utilizado (Classe 700).....	38
Figura 21 - Quadricóptero utilizado	39
Figura 22 - Montagem dos dispositivos que irão embarcados.....	40
Figura 23 - Placa FTDI e modem de comunicação	41
Figura 24 - Aplicação de Vant em fotografia aérea	44
Figura 25 - Câmera GoPro Hero3	45
Figura 26 - Câmera FPV CCD	45
Figura 27 - Vídeo Rx, Tx.....	46
Figura 28 - Antena <i>patch</i> de 8dbi.....	46
Figura 29 - Placa de Captura USB	47
Figura 30 - Sistema Pan-Tilt com dois Servos.....	47
Figura 31 - Comandos para instalação do reprodutor de mídia	48
Figura 32 - Comandos para adicionar o vídeo à GCS.....	48
Figura 33 - Comandos para posicionar o vídeo no GCS	49
Figura 34 - Exemplo de RNA: A) Arquitetura e B) Neurônio.....	51
Figura 35 - Imagem térmica de um transformador.....	52
Figura 36 - Teste SOM, imagem original de entrada	57
Figura 37 - Teste SOM, imagem após a aplicação do passo um.....	58
Figura 38 - Teste SOM, imagem após o passo dois	58
Figura 39 - Teste SOM, imagem de saída.....	59

Lista de Tabelas

Tabela 1 - Uma classificação por categoria, peso e alcance dos VANTs.....	12
Tabela 2 - Padrões de temperatura projetados para testar a eficiência da MLP	53
Tabela 3 - Atributos utilizados nos testes feitos com o SOM Kohonen.....	57

Siglas

- AI - *Artificial Intelligence* (Inteligência Artificial - IA)
- AIC - *Aeronautical Information Circular* (Circular de Informação Aeronáutica)
- ANAC - Agência Nacional de Aviação Civil
- ANN - *Artificial Neural Network* (Rede Neural Artificial - RNA)
- CCD - *Charge-Coupled Device* (Dispositivo de Carga Acoplada)
- CMOS - *Complementary Metal Oxide Semiconductor* (Semicondutor de Metal Óxido

Complementar)

- DECEA - Departamento de Controle do Espaço Aéreo
- FPV - *First Person View* (Visão em Primeira Pessoa)
- GCS - *Ground Control Station* (Estação de Controle em Terra)
- GPS - *Global Position System* (Sistema de Posicionamento Global)
- GUI - *Graphical User Interface* (Interface Gráfica do Usuário)
- HD - *High Definition* (Alta Definição)
- IA - Inteligência Artificial
- IMU - *Inertial Measurement Unit* (Unidade de Medição Inercial)
- MLP - *Multilayer Perceptron* (*Perceptron* Multicamadas)
- PCB - *Printed Circuit Board* (Placa de Circuito Impresso)
- PDI - Processamento Digital de Imagens
- PID - *Proportional Integral Derivative* (Proporcional Integral Derivativo)
- RNA - Rede Neural Artificial
- RPAS - *Remotely Piloted Aircraft System* (Sistema de Aeronaves Remotamente

Pilotadas)

- RX - Receiver (Receptor)
- SOM - *Self Organizing Map* (Mapa Auto-Organizável)
- TX - *Transmitter* (Transmissor)
- UAV - *Unmanned Aerial Vehicle* (Veículo Aéreo Não Tripulado)
- USB - *Universal Serial Bus* (Barramento Serial Universal)
- VANT - Veículo Aéreo Não Tripulado

1 Introdução

Nos dias atuais, tem surgido muito interesse em torno de Veículos Aéreos Não Tripulados (VANTs), especialmente no âmbito de coleta de informações. Inicialmente, seu uso foi nas forças armadas, mas, recentemente, a área civil tem aproveitado bastante dos benefícios que estes podem trazer, principalmente pelo baixo custo envolvido.

É muito comum se encontrarem áreas restritas ao redor do planeta, as quais necessitam de monitoramento, de modo a garantir sua integridade, seja evitando invasões, desmatamentos ilegais e alterações no terreno, entre outros motivos. Os VANTs podem viajar distâncias consideráveis com segurança a um custo muito menor do que outros métodos tradicionais para inspeção, como alternativa a utilização de usando helicópteros tripulados quando o custo do combustível já seria muito maior. Além desses fatores, pode-se citar a fadiga e o cansaço de tripulações devido a extensas horas de trabalhos, os quais podem atrapalhar as operações.

Uma aeronave autônoma pode capturar imagens de diversos ângulos, que podem ser processadas por Rede Neural Artificial (RNA), a fim de identificar problemas encontrados dentro da área definida.

As RNAs são técnicas de modelagem de Inteligência Artificial (IA), que tentam emular as habilidades computacionais do cérebro humano, onde o conhecimento é mantido através de conexões entre neurônios, chamados de sinapses. Por exemplo, o Mapa Auto-organizável Kohonen (SOM Kohonen) é um tipo de rede neural competitiva com aprendizagem não supervisionada (HAYKIN, 1999). Nele, para cada sinal de entrada, apenas o neurônio com maior afinidade é ativado. Isto é, se as entradas da rede neural são os *pixels* de uma imagem, para cada *pixel*, somente um neurônio é ativado, configurando, assim, um grupo ou classe ao qual aquele *pixel* pertence. No final do processo, há uma nova imagem de saída com os *pixels* agrupados de acordo com características similares.

1.1 Classificação

Uma aeronave não tripulada pode ter sua classificação baseada em 16 tipos diferentes conforme apontam Korchenko e Illyash (2013). São eles:

- 1) aplicação da aeronave;
- 2) tipo de sistema de controle;
- 3) regra de voo;
- 4) classificação do espaço aéreo a ser utilizado (nacional ou internacional, por exemplo);
- 5) tipo de aeronave;
- 6) tipo de asa;
- 7) direção de decolagem / pouso;
- 8) tipo de decolagem / pouso;
- 9) tipo de motor da aeronave;
- 10) sistema de combustível;
- 11) tipo de tanque de combustível;
- 12) número de explorações;
- 13) categoria (de acordo com o peso e distância máxima de voo);
- 14) raio de voo;
- 15) altitude de voo; e
- 16) funções da aeronave.

No escopo deste trabalho, o tipo de classificação considerado será categoria, encontrado no item 13. É fundamental que se utilizem veículos aéreos de baixo custo, com o intuito de se reduzirem as aplicações pretendidas e as despesas envolvidas com o monitoramento aéreo. Mantendo esse pensamento, pode-se observar, na Tabela 1, uma classificação de aeronaves por categoria:

Tabela 1 – Uma classificação por categoria, peso e alcance dos VANTs

Categoria	Subcategoria	Massa (kg)	Alcance Máximo de Ação (km)	
			Com retorno	Sem retorno
Tático (Voos de baixo alcance)	Nano, η	0,025	menos de 1	
	Micro, μ	menos de 5	menos de 10	menos de 20
	Mini	menos de 25	menos de 30	menos de 60
	Close Range (CR)	25-150	10-30	30-80
Operativo-Tático (Voos de baixo alcance)	Short Range (SR)	50-250	30-80	90-160
Operativo (Voos de alcance mediano)	Medium Range (MR)	150-500	80-200	240-400
	Medium Range Endurance (MRE)	500-1500	200-500	600-1000
	Low Altitude Deep Penetration (LADP)	250-2500	250-300	700-800
Operativo-Estratégico (Voos de longa duração)	Low Altitude Long Endurance (LALE)	150-250	500-800	1500-1600
	Medium Altitude Long Endurance (MALE)	1000-1500	500-800	1500-1600
Estratégico	High Altitude Long Endurance (HALE)	2500-5000	menos de 2500	menos de 2000
Especial	Unmanned Combat Aerial Vehicle (UCAV)	mais de 1000	não mais do que 1500	não mais do que 1500
	Lethal (LETH)			
	Decoy (DEC)			
	Stratospheric (STRATO)			
	Exostratospheric (EXO)			
	Spatial (SPACE)			

Fonte: Korchenko e Illyash (2013).

Na Tabela 1, observa-se que existem VANTs de vários tamanhos e pesos. Todavia, para que o trabalho continue no foco de baixo custo, testes ocorreram com a Categoria de VANT Tático considerada de baixo alcance, principalmente os das Subcategorias Micro μ e Mini, por possuírem tamanho reduzido, estrutura fácil de ser construída ou adquirida e capacidade de carga mínima o suficiente para levantar voo com dispositivos de gravação e controle de navegação acoplados.

1.2 Regulamentação no Brasil

No Brasil, dois órgãos são importantes para exercer o controle aéreo dos VANTs e seus equipamentos: o Departamento de Controle do Espaço Aéreo (DECEA), a quem compete normalizar e controlar o uso e o acesso ao espaço aéreo e a segurança (*safety*) de navegação aérea. Já as questões de certificação de equipamentos e licenças de pilotos cabem à Agência Nacional de Aviação Civil (ANAC).

Em setembro de 2010, o DECEA emitiu a Circular de Informações Aeronáuticas (AIC-N 21/10), que continha, na ocasião, a legislação sobre os VANTs, ou Aeronaves Remotamente Pilotadas (RPAS), como são, normalmente, nomeadas, e também seu acesso ao espaço aéreo brasileiro. Já em 19 de novembro de 2015, a publicação ICA 100-40 foi emitida legislando os VANTs com o intuito de flexibilizar ao usuário o acesso ao espaço aéreo.

Além dessas publicações, uma Portaria também foi emitida pelo Comando da Aeronáutica (1999), tratando do uso de aeromodelos e das seguintes regras importantes de segurança (*safety*):

a) A operação de aeromodelos deve ser realizada em locais suficientemente distantes de áreas densamente povoadas. Deve ser evitada a operação de aeromodelos motorizados nas proximidades de áreas ou instalações urbanas sensíveis ao ruído como hospitais, templos religiosos, escolas e asilos;

b) Deve ser evitada a operação de aeromodelos na presença de público até que estes sejam testados em voo, com êxito, e se comprove segurança (*safety*) nas suas operações;

c) A menos que autorizado, nenhum aeromodelo deve ser operado a mais de 400 pés acima da superfície terrestre. A operação de aeromodelos nas proximidades de aeródromos somente poderá ser executada após autorização do responsável pela operação do aeródromo;

d) É proibida a operação de aeromodelos nas zonas de aproximação e decolagem dos aeródromos;

e) As operações, com equipamentos rádio controlados, distintas de esporte e lazer deveriam ser submetidas ao Departamento de Aviação Civil, extinto em 2006 e hoje absorvido pela ANAC; e

f) Em caso de dúvidas, procure a ANAC (COMANDO DA AERONÁUTICA, 1999).

1.3 Objetivos

O objetivo deste trabalho se define em configurar um Veículo Aéreo Não Tripulado (VANT), envolvendo os aspectos relacionados com a utilização dos VANTs táticos em voos de baixo alcance capazes de realizar missões do tipo vistorias em linhas de transmissão de energia, inspeções aéreas em busca de desmatamentos ilegais e mapeamento urbano. Para isto, uma aeronave de asa fixa ou rotativa deve ser configurada de forma a realizar voos autônomos, facilitando o uso para seu operador, que somente precisará posicionar o aeromodelo no ponto inicial de voo. A partir daí, ele realizará operações de decolagem e pouso automáticos, filmando ou tirando fotos de todo o local previamente estabelecido.

O processo tradicional de inspeção em linhas de transmissão, por exemplo, é caro e requer treinamentos rigorosos para evitar riscos de acidentes. Entre os objetivos deste trabalho, estão a diminuição nos custos de tais operações e, principalmente, a redução do risco de acidentes sem perder qualidade e a eficácia. A Figura 1 mostra um exemplo de inspeção tradicional em uma torre de transmissão utilizando um helicóptero de tamanho real juntamente com sua tripulação, que, algumas vezes, arriscam suas vidas subindo na linha energizada.



Figura 1 - Exemplo de inspeção usando o método tradicional
Fonte: (JACKSON, 2009)

1.4 Organização do Trabalho

A divisão deste trabalho é feita de forma a facilitar sua leitura e será feita em capítulos numerados conforme sumário. Alguns termos utilizados neste trabalho de pesquisa encontram-se em Inglês por terem sua definição mais clara e não possuírem tradução em Português.

O Capítulo 2 trata da teorização dos elementos envolvidos, quando discute-se o funcionamento dos sistemas aéreos não tripulados e algumas possíveis opções existentes no mercado. Em seguida, detalham-se, respectivamente, o *hardware* e o *software* do sistema escolhido finalizando com uma introdução as técnicas de inteligência artificial utilizadas.

A proposta de solução e sua aplicação encontram-se descritas, respectivamente, nos Capítulos 3 e 4. O Capítulo 3 descreve as aeronaves utilizadas neste trabalho de pesquisa, a construção e a programação utilizada para interligar os sistemas eletrônicos nos VANTs, os problemas e soluções em gravações e tomada de fotos utilizando veículos aéreos não tripulados. Por fim, o Capítulo 4 descreve as aplicações desenvolvidas durante os estudos.

O Capítulo 5 sintetiza a conclusão do trabalho juntamente com possíveis direcionamentos para trabalhos futuros.

2 Fundamentação Teórica

Neste capítulo, encontram-se os fundamentos necessários para o entendimento do problema e as explicações das escolhas para uma solução proposta. A primeira seção trata da teoria dos sistemas aéreos não tripulados (AUSTIN, 2010) e detalha alguns dos sistemas mais utilizados nessa área.

Em seguida, descreve-se o sistema proposto, envolvendo o *hardware* do próprio piloto automático, os microcontroladores, a unidade de medida inercial, o sistema de posicionamento global, o *modem* de comunicação e, finalizando, a parte de *hardware* com o rádio e conversor de sinais. Além disso, este capítulo propicia uma visão do *software* do sistema, abrangendo a central de comandos e a estação de controle (GCS).

2.1 Sistemas Aéreos Não Tripulados

Um sistema aéreo não tripulado compreende uma série de subsistemas que incluem a aeronave (muitas vezes referida como UAV – *Unmanned Aerial Vehicle* – ou, em português, VANT – Veículo Aéreo Não Tripulado), sua carga, a estação de controle e os subsistemas de comunicação, entre outros (AUSTIN, 2010).

Os VANTs são definidos como aeronaves que não carregam uma tripulação, podendo ser controlados remotamente ou voar de maneira autônoma. Neste caso, o termo UAS (*Unmanned Aerial System*) pode substituir corretamente a sigla VANT (BEHNCK, 2014).

Este trabalho de pesquisa trata exclusivamente de sistemas autônomos para reconhecimento de imagens e mudanças de atitudes na tomada de decisão e será utilizado um dos vários UAS disponíveis. A seguir, serão mostradas algumas opções encontradas e o sistema escolhido.

2.1.1 ArduPilot

Um dos sistemas aéreos não tripulados mais utilizados em VANTs táticos e com uma grande comunidade que desenvolve sistemas abertos (*Open Source*) é o Ardupilot, um piloto automático repleto de recursos e tem como base de processamento e aquisição de sinais a plataforma Arduino. O Ardupilot é uma placa de circuito impressa, personalizada com um processador embutido juntamente com circuitos, para alterar entre o controle pelo rádio ou autônomo. No seu segundo modo, autônomo, ele é capaz de propiciar a navegação pontos dados por meio do GPS. Este é um sistema totalmente programável e seu desenvolvimento é baseado em sistemas *Open Source* (AMAHAH, 2009).

Algumas de suas características incluem um *hardware* embutido, à prova de falhas, que se utiliza de um circuito separado para transferir o controle do modo automático para o modo de controle por rádio, e vice-versa. Ele possui a capacidade de reiniciar o processador principal em pleno voo além de prover a possibilidade de utilização de diversos *waypoints*, assim como uma conexão para um GPS (BIN; JUSTICE, 2009).

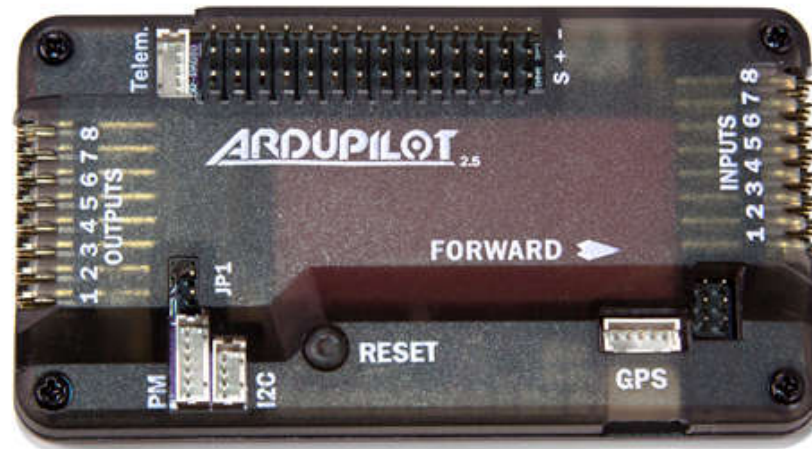


Figura 2 - Uma Placa Ardupilot 2.5 Disponível no Mercado
Fonte: (ARDUPILOT, 2016)

A Figura 2 mostra a imagem de uma placa Ardupilot, uma das construções físicas disponíveis. Já a Figura 3 mostra a interface da estação de trabalho.

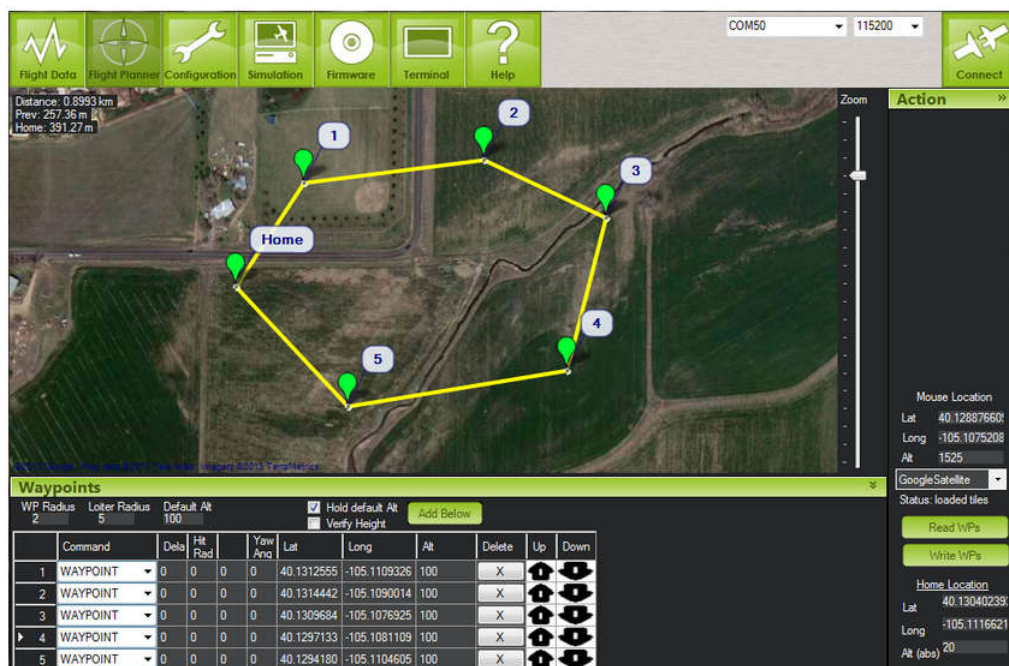


Figura 3 - Um Exemplo de Interface do Sistema Ardupilot
Fonte: (ARDUPILOT, 2016)

2.1.2 Paparazzi

O sistema denominado Paparazzi teve início do seu desenvolvimento em 2003, como um projeto em forma de passatempo. Após sua participação bem-sucedida em vários torneios, este sistema se tornou amigável ao usuário o suficiente para grupos de pesquisa com interesses científicos variados. Este foi o primeiro projeto de sistema de autopiloto a abordar a ideologia de código fonte aberto (*Open Source*). Como resultado desta abordagem, um grande número de laboratórios de pesquisa, instituições e universidades aderiram a este projeto de sistema até o momento (GATI, 2013).

Este sistema já foi usado com sucesso em diversas aplicações, por exemplo: inspeção do vento em três dimensões em um parque eólico na Suíça (MANSOUR *et al.*, 2010), tomada de 8.500 imagens multiespectrais, a fim de melhorar o ecossistema em uma das regiões mais pobres do mundo em Madagascar e, até mesmo, na Antártica, onde uma equipe o utilizou em uma aeronave de asa fixa, em mais de 90 vezes, durante duas semanas, para realizar medições meteorológicas (KNUTH *et al.*, 2013).

O Paparazzi foi o escolhido para este trabalho por ser um sistema mais voltado à pesquisa e ter características como a utilização de licença GNU (*General Public License*). Nas seções 2.2 e 2.3, será possível encontrar em mais detalhes as suas funcionalidade e aplicabilidades.

2.1.3 Outros Sistemas

Além dos dois sistemas citados anteriormente, existem outros, alguns de característica comercial e outros também de licença livres. Como exemplo, pode-se citar o MicroPilot, OpenPilot, wePilot™, Vector e vários outros existentes no mercado. Esses sistemas não serão detalhados neste trabalho, mas são opções a serem consideradas e cada uma delas possuem suas características, com vantagens e desvantagens que dependem da aplicação.

2.2 O *Hardware* do Sistema Computadorizado Escolhido

Definido o sistema *Open Source* Paparazzi como o autopiloto, descrever-se-á sobre o *hardware* e o *software* envolvidos neste sistema, a começar com o *hardware*.

2.2.1 Autopiloto e Microcontroladores

Uma das características do Paparazzi é o suporte para projetos com vários tipos diferentes de *hardware*. A placa microcontroladora escolhida para o desenvolvimento deste trabalho foi a placa denominada Lisa/M e os motivos para a escolha serão mencionados a seguir.

Lisa/M é uma placa miniatura de piloto automático, de uso geral e flexível entre múltiplas plataformas. De pequeno peso e tamanho, pode possuir integrada a ela a placa IMU Aspirin. O tamanho e especificações adequados fazem com que a Lisa/M sirva para os veículos de asa fixa e de asa rotativa. Esse piloto automático baseia-se na melhoria da STM32 para periféricos e um processador muito rápido.

Lisa/M é baseada no processador de 64 pinos STM32F105RCT6 com 64k de memória RAM e 256k de FLASH. Todos os pinos são expostos, fornecendo acesso ao conjunto completo de periféricos STM32. Suas características mais completas são listadas a seguir:

- Microcontrolador STM32 do tipo STM32F105RCT6. Ficha técnica com 256kB *flash* e 64kB RAM;
- Sensor de pressão BMP085;
- sete Canais de entrada analógicos;
- três Entradas/Saídas digitais genéricas;
- dois 3.3V TTL UART (5V);
- oito Saídas Servo PPM;
- um Barramento CAN;
- um Barramento SPI;
- um Barramento I2C;
- um Micro USB;
- quatro *status* LEDs com teste de ponto integrado;
- 10,8 gramas (com a IMU Aspirin montada);
- 9,9 gramas (sem a IMU Aspirin montada);
- Dimensões de 34 mm x 60 mm x 10 mm;

- Placa de circuito impresso (PCB) de quatro camadas (GATI, 2013).

A Figura 4 representa a placa Lisa/M V2.0 utilizada no projeto (vista superior) com seus respectivos pinos de conexão para montagem elétrica.

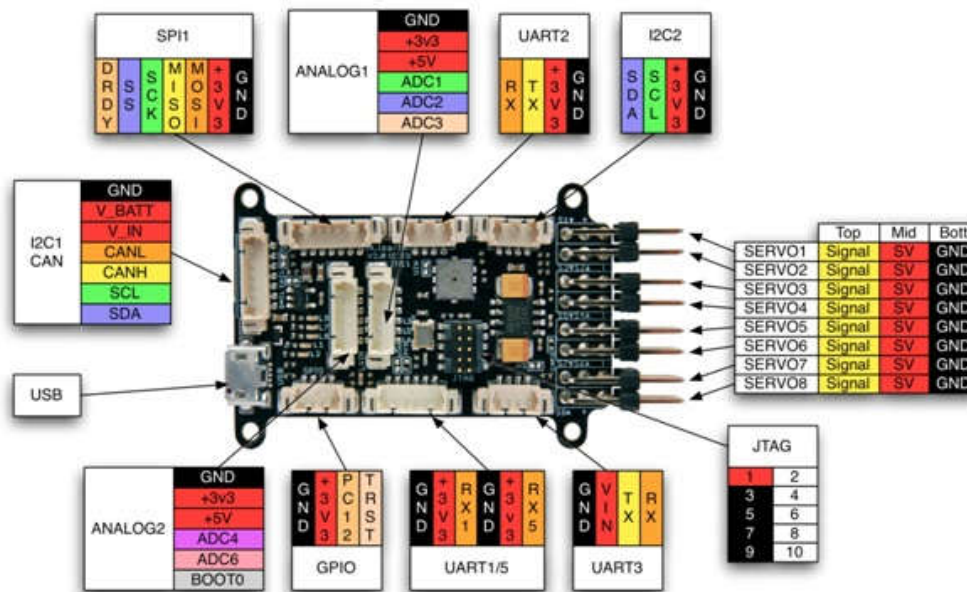


Figura 4 - Lisa/M V2.0, visão por cima com pinagem

Fonte: (PAPARAZZI, 2016)

Na Figura 5, tem-se a placa Lisa/M V2.0 (vista inferior), na qual é possível encontrar seu processador, outros circuitos integrados auxiliares e a descrição dos pinos e conexões.

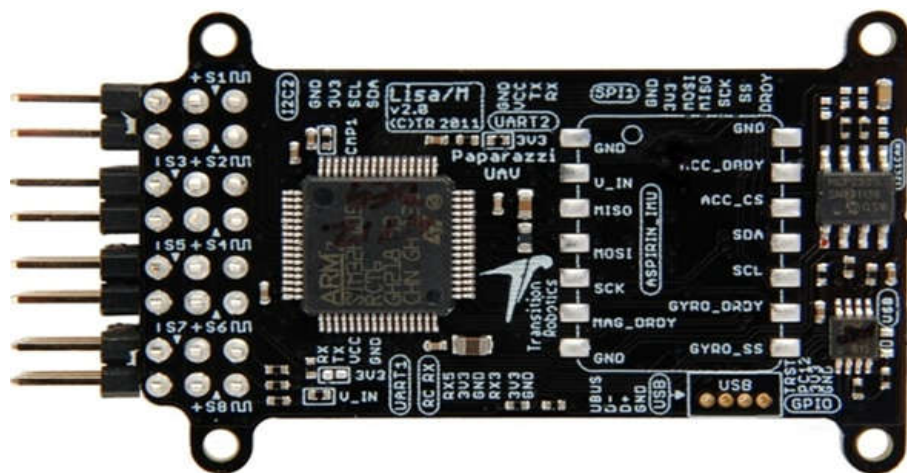


Figura 5 - Lisa/M V2.0, visão por baixo

Fonte: (PAPARAZZI, 2016)

2.2.2 Unidade de Medida Inercial (IMU)

A Unidade de Medida Inercial, do inglês *Inertial Measurement Unit* (IMU), é usada para medir as acelerações, rotações angulares e, em alguns casos, o campo magnético da Terra. A IMU utilizada neste trabalho foi a Aspirin IMU V2.1 por possuir compatibilidade e sincronia perfeita com a placa microcontroladora escolhida: Lisa/M.

Considerada a nova geração de unidade de medida inercial, a Aspirin IMU usa sensores digitais de três eixos e possui um tamanho reduzido e de baixo custo. Outras características incluem reguladores de tensão *onboard*, EEPROMs para armazenamento de dados de calibração e sensores extras, como, por exemplo, um barômetro.

A seguir, listam-se as principais características da Aspirin IMU:

- Acelerômetro de três eixos;
- Giroscópio de três eixos;
- Magnetômetro de três eixos;
- Regulação de voltagem *on-board*;
- EEPROM I2C *on-board*;
- Barômetro *on-board*;
- *Footprint* padrão entre todas as versões;
- PCB *flat form-factor* único; e
- Encaixe e pinagem compatível para a Lisa/M.

As Figuras 6 e 7 mostram, respectivamente, as visões superior e inferior da placa IMU utilizada neste trabalho.

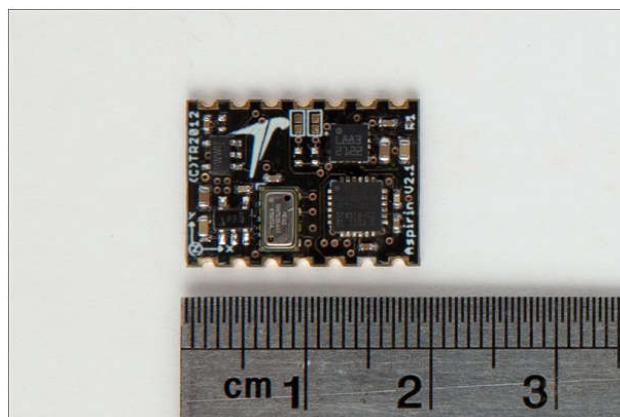


Figura 6 - Aspirin IMU, visão de cima
Fonte: (PAPARAZZI, 2016)



Figura 8 - GPS U-blox LEA 5S
Fonte: autoria própria.

2.2.4 Modems de Comunicação

Dois *modems* para comunicação entre o computador – mais tarde, chamado de estação de controle em terra (GCS) – e o aeromodelo foram necessários. Um *link* bidirecional entre esses dois *modems* permite que as informações de telemetria cheguem à tela da GCS em tempo real. Além da telemetria, outras funções são viáveis com essa comunicação, por exemplo: o ajuste fino de variáveis na aeronave, assim como comandos de navegação e mudanças de pontos de passagem (*waypoints*) durante voo. O sistema também pode, com os *modems*, se comunicar e fazer um voo com várias aeronaves em formação. Porém, para isso, são necessários mais de dois *modems*. Esta possibilidade de controle de voo não foi testada neste trabalho por não pertencer ao escopo do mesmo.



Figura 9 - Modem XBee Pro ZB com saída pigtail
Fonte: autoria própria.

Os *modems* adquiridos foram XBee Pro ZB. Tais *modems* possuem comunicação na frequência de 2.4 GHz e, por isso, são totalmente permitidos pela legislação brasileira, além de não sofrerem interferência com outros *modems* de mesma frequência. Um deles tem uma saída *pigtail* para o usuário colocar a antena de sua preferência. Um desses *modems* vai na GCS juntamente com um sistema para fazer a comunicação com a USB do computador. O segundo *modem* possui um antena embutida muito pequena para poder ser carregado sem problemas em qualquer que seja a aeronave empregada.

A Figura 9 mostra o primeiro *modem* com a saída *pigtail* e o segundo é mostrado na Figura 10 com sua própria antena embutida.



Figura 10 - Modem XBee Pro ZB com antena pequena

Fonte: autoria própria.

2.2.5 Rádio, Receptor RC e Conversor de Sinais

Para ser possível o experimento e testar todos os níveis do *software* e do *hardware* com uma chance reduzida de acidentes, deve-se ter a possibilidade de trocar entre os comandos automáticos, semiautomáticos e totalmente manuais. Esse último, para o caso de alguma falha, um piloto experiente poderia trocar de posição uma chave no rádio, correspondente à seleção do modo de comando, e assumir totalmente o controle da aeronave. Portanto, um

rádio com uma chave de três posições se fez necessário. Respectivamente, o rádio e o receptor utilizado são mostrados nas Figuras 11 e 12.



Figura 11 - Rádio com chave de três posições
Fonte: autoria própria.



Figura 12 - Receptor do rádio
Fonte: autoria própria.

Além do receptor, o rádio foi precisou também uma placa transformadora de sinais. Tal placa recebe o sinal conhecido como modulação por posição de pulso, ou do Inglês *Pulse-Position Modulation* (PPM). Este transforma os vários sinais recebidos pelo receptor em um único sinal de saída necessário para que o *software* entenda os comandos. A Figura 13 mostra o decodificador conhecido como ATmega PPM Encoder usado no trabalho.



Figura 13 - Placa ATmega PPM Encoder
Fonte: autoria própria.

2.3 O *Software* do Sistema Computadorizado Escolhido

Esta seção descreve a capacidade e o detalhamento do *software* envolvido na realização das configurações do sistema: duas telas mais importantes, ou seja, a central paparazzi e a estação de controle de terra (GCS), ambas descritas a seguir.

2.3.1 Central Paparazzi

A Central Paparazzi é uma interface gráfica que contém três páginas principais:

- um conjunto de caixas de seleção para configurar uma aeronave com o seu plano de voo e construir os programas correspondentes a serem simulados e enviados para o dispositivo aéreo;
- um painel de controle para iniciar os agentes do sistema e lidar com uma coleção de programas, como sessões configuráveis; e
- uma página na qual o GCS pode ser incorporado.

Um console de *log* também é exibido nas páginas do painel de configuração e controle.

A Figura 14 ilustra um exemplo da tela da Central Paparazzi. A seguir, são descritos os diversos subitens desta tela.

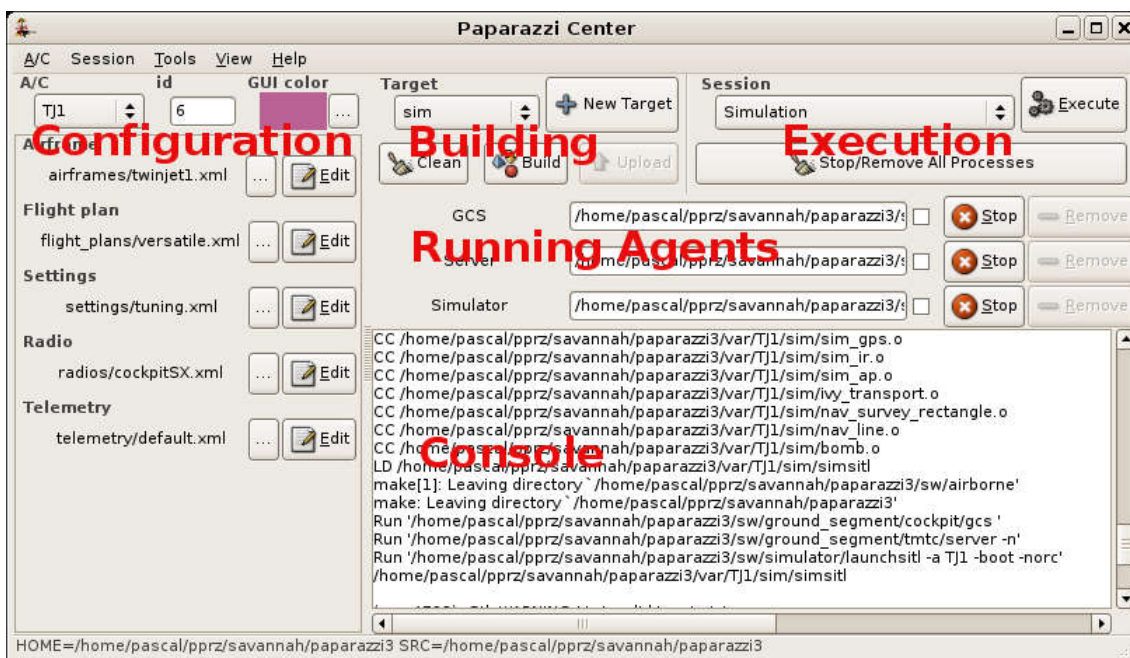


Figura 14 - Paparazzi Center
Fonte: (PAPARAZZI, 2016)

Configuração

A parte esquerda da página de configuração é um editor para o arquivo de configuração `conf.xml`. Uma nova aeronave pode ser acrescentada a partir do menu A/C. A aeronave atual pode ser excluída do menu A/C.

O editor para os arquivos de configuração é retirado da variável de ambiente EDITOR.

Vários arquivos de configuração podem ser simultaneamente selecionados para o atributo de configuração usando a tecla CTRL no seletor de arquivo.

Compilação

A compilação e a gravação são feitas a partir de um painel de construção. Os alvos podem ser adicionados à *combo box* com o botão novo alvo. Compilação e comandos de gravação são mostrados como agentes de execução para que eles possam ser interrompidos, se necessário.

Execução

No painel de Execução, uma *combo box* fornece um conjunto predefinido de sessões.

A sessão de simulação é um servidor, um GCS e um simulador para a aeronave selecionada no painel de configuração. Nota-se que o alvo *sim* deve ter sido construído antes da execução da simulação. Ou seja, deve-se passar as etapas de compilação previamente para que seja realizada a execução de uma simulação.

Os programas executados podem ser parados e reiniciados (botões Parar/Refazer). As opções também podem ser editadas na caixa de entrada. A partir do *menu* de sessão, os conjuntos dos processos atuais podem ser salvos como uma sessão para serem reiniciados mais tarde.

2.3.2 Estação de Controle em Terra (GCS)

A Estação de Controle em Terra, do Inglês *Ground Control Station* (GCS), é um operador de controle de unidade de *software* para o controle de veículos aéreos. Ela permite visualizar e controlar um veículo aéreo durante o desenvolvimento e a operação, tanto em ambiente *indoor* quanto em ambiente *outdoor*, com uma arquitetura de *software* flexível que suporta vários tipos Micro UAV e projetos de piloto automático. O objetivo do GCS é o monitoramento em tempo real de um UAS.

O Paparazzi GCS é uma aplicação rica em recursos com interfaces gráficas totalmente personalizáveis, cada uma contendo uma coleção de componentes de interface mais úteis para um propósito particular.

Dentre suas características, estão:

- voo com suporte para várias aeronaves simultâneas;
- suporte multissistemas (protocolos múltiplos, pilotos automáticos múltiplos e projetos múltiplos);
- mapa 2D capaz de exibir imagens do *Google Satellite*, *OpenStreetMaps* e *Microsoft Satellite Maps*;
- planejamento de missão;
- movimentação de *waypoints* em tempo real;
- ajustes em tempo real dos planos de voo, se necessário;
- visão geral do *status* do sistema;
- calibração da aeronave em tempo real mesmo estando no ar;
- suporta aeronaves de asas rotativas e de asas fixas como, por exemplo, aviões, helicópteros coaxiais, e quadricópteros;
- teclas de atalho definíveis para acesso rápido e simples de determinados comandos em campo; e
- GUI totalmente configurável à disposição do usuário.

Na Figura 15, apresenta-se uma amostra da tela de um dos voos realizados por uma das aeronaves, que percorreu, de modo totalmente autônomo, uma trajetória na forma oval.

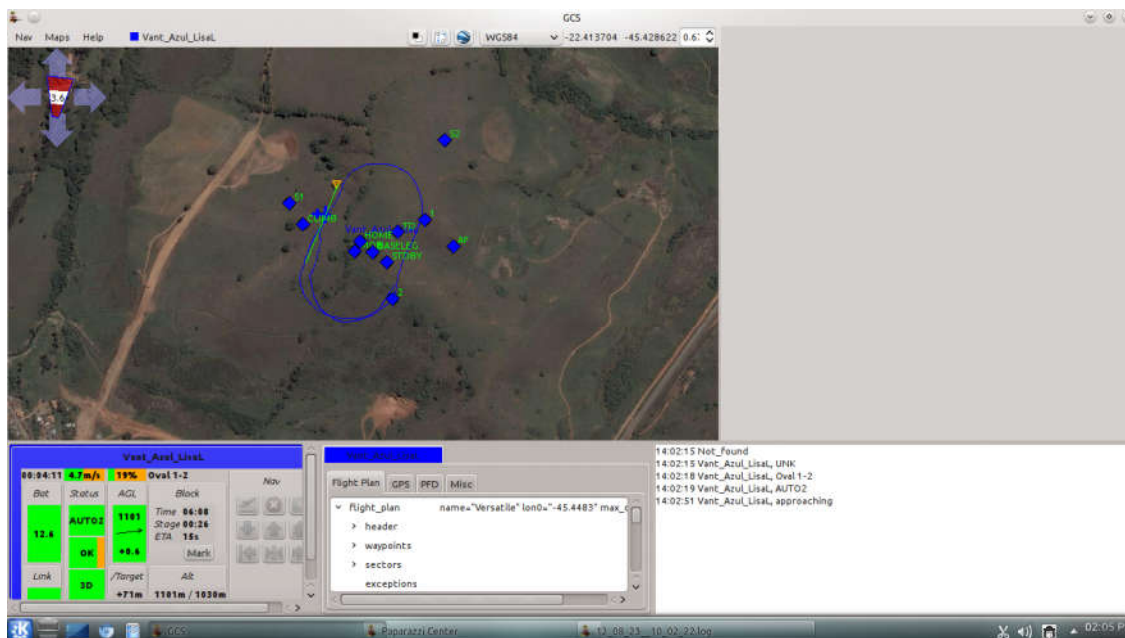


Figura 15 - Trajetória oval mostrada na GCS
 Fonte: autoria própria.

Nesta seção, foram mostrados os aspectos do *software* utilizado no trabalho. No capítulo 3, serão tratadas as questões do desenvolvimento e escolha das tecnologias.

2.4 Inteligência Artificial e Redes Neurais Artificiais

No capítulo 4 desta pesquisa serão apresentados dois estudos de casos realizados com aplicações de VANTs autônomos de baixo custo. Para o desenvolvimento destes estudos de casos foram utilizadas técnicas de Inteligência Artificial (IA), mais especificamente Redes Neurais Artificiais (RNA) que serão apresentadas nesta seção.

As redes neurais artificiais também são utilizadas em processamento de imagens. De forma geral, uma rede neural artificial é uma máquina projetada para modelar a maneira como o cérebro realiza uma tarefa particular ou função de interesse; é normalmente implementada utilizando-se componentes eletrônicos ou é simulada por programação em um computador digital (HAYKIN, 2001).

Para conseguir um grande desempenho, as redes neurais artificiais utilizam um grande número de interligações entre células computacionais simples chamadas de “neurônios”. Dessa forma, pode-se utilizar a seguinte definição de uma RNA vista como uma máquina adaptativa.

Uma RNA é um processador maciço e paralelamente distribuído constituído de unidades de processamento simples, que têm a propensão natural para armazenar conhecimento experimental e torná-lo disponível para o uso. Ela se assemelha ao cérebro em dois aspectos: o conhecimento é adquirido pela RNA a partir de seu ambiente através de um processo de aprendizagem; e forças de conexão entre os neurônios são utilizadas para armazenar conhecimento adquirido.

O aprendizado é realizado através do algoritmo de aprendizagem, cuja função é modificar os pesos sinápticos da rede neural (as forças de conexão) de uma forma ordenada para aproximar a resposta da RNA do objetivo do projeto desejado. Os modelos neurais computacionais, as RNAs, procuram aproximar o processamento dos computadores ao cérebro humano.

As RNAs extraem seu poder computacional de sua estrutura e de sua habilidade de aprender e generalizar. A generalização se refere ao fato da RNA produzir saídas adequadas para valores que não foram apresentados anteriormente a ela. Devido a essas capacidades de processamento de informação é que as RNAs podem resolver problemas complexos (MOTA, 2014).

O neurônio artificial é um dispositivo lógico-matemático que procura simular as funções e o comportamento de um neurônio biológico. Assim sendo, os dendritos são substituídos por entradas, onde ligações com o corpo celular artificial são realizadas através de elementos

denominados pesos, que simulam as sinapses. Os estímulos captados pelas entradas são processados pela função de soma ponderada dos sinais, e o limiar de disparo do neurônio biológico é substituído pela função de transferência (HAYKIN, 2001).

Com a combinação de diversos neurônios artificiais, forma-se a chamada RNA. A partir de uma RNA formada, uma série de valores pode ser aplicada sobre um neurônio, sendo que este está conectado a outros pela rede neural. Estes valores, ou entradas, são multiplicados no neurônio pelo valor do peso de sua sinapse. Posteriormente, esses valores são somados. Se esta soma ultrapassar um valor limite estabelecido, um sinal é propagado pela saída, axônio, deste neurônio. Em seguida, essa mesma etapa se realiza com os demais neurônios da rede neural. Isso quer dizer que os neurônios vão enfrentar algum tipo de ativação, dependendo das entradas e dos pesos sinápticos (HAYKIN, 2001).

Existem várias formas de se desenvolver uma rede neural. Ela deve ser montada de acordo com os problemas a serem resolvidos, portanto deve-se determinar a arquitetura da RNA. Em sua arquitetura é determinado o número de camadas usadas (formadas por neurônios) e a quantidade de neurônios em cada camada. Existem dois tipos principais de redes neurais: as supervisionadas e as não supervisionadas (HAYKIN, 2001). As supervisionadas classificam padrões de acordo com seu aprendizado e se obtém uma resposta de acordo com esse aprendizado.

Como exemplos, têm-se as redes MLP e o perceptron de camada única; e não supervisionadas classificam padrões sem um aprendizado prévio, de acordo com a similaridade entre os padrões, tendo como principal exemplo o mapa auto-organizável de Kohonen (KOHONEN, 2000).

3 Integração de *Software* e *Hardware*

Este trabalho de pesquisa é considerado de natureza tecnológica, com uma abordagem qualitativa e objetivos exploratórios. Está classificado como pesquisa-ação, utilizando técnicas de simulação e o desenvolvimento de protótipos reais. Ele foi realizado em campo e em laboratório (GIL, 1946).

A execução deste trabalho foi realizada nos âmbitos do Laboratório de Multimídia e Interatividade, situado nas dependências do Departamento de Matemática e Computação (DMC), da Universidade Federal de Itajubá (UNIFEI), e se deu por um período de três anos completos, entre janeiro de 2013 e janeiro de 2016.

Este capítulo descreve como se deu o desenvolvimento do trabalho, desde a escolha dos modelos de asa fixa até os de asa rotativa, bem como as configurações de *hardware*, *software* e vídeo utilizadas.

3.1 Aeronaves

Uma aeronave é qualquer máquina que seja capaz de voar sendo sustentada no ar por meio de flutuabilidade ou forças aerodinâmicas (COLLINS DICTIONARY, 2014). Portanto, essa é uma categoria muito extensa, que pode envolver balões, planadores, helicópteros, multicópteros, aviões, motoplanadores e outros.

Para o contexto deste trabalho, serão destacadas e utilizadas as aeronaves de asa fixa, como aviões, e também de asa rotativa, como quadricópteros e helicópteros. Esses são os modelos mais usados no âmbito do aeromodelismo juntamente com aeronaves que batem asas, chamadas de ornitópteros, que não serão tratadas neste trabalho.

3.1.1 Asa Fixa

As aeronaves de asa fixa têm seu representante mais comum o avião, e é esse o primeiro aeromodelo que será empregado no projeto. Essas aeronaves podem ser impulsionadas por motores a jato ou hélices. Sua origem é ainda bem discutida até hoje por profissionais e entidades da área com uma histórica disputa entre os irmãos Wright (1903) e Santos-Dumont (1906).

Basicamente, dois aeromodelos em formato de avião foram usados e são mostrados nas Figuras 16 e 17.



Figura 16 - Primeiro aeromodelo de asa fixa (monomotor)

Fonte: autoria própria.

O aeromodelo monomotor utilizado foi de simples construção, porém com uma ótima estabilidade e, ao mesmo tempo, robusto, caso haja algum tipo de acidente ou queda. O modelo foi projetado e construído por um dos professores colaboradores do projeto, Prof. Dr. Marcos Theiss Neves, modelo desenhado especificamente para iniciantes na prática do aeromodelismo. Por isso, possuía todas as características necessárias para iniciar as experiências.



Figura 17 - Segundo aeromodelo de asa fixa (bimotor)

Fonte: autoria própria.

O segundo aeromodelo utilizado foi adquirido com o propósito de se ter uma aeronave maior, mais sofisticada e que possuísse dois motores, pois, no caso de uma falha em um dos motores, ainda poderia voar com o outro motor funcionando. Vale ressaltar que o primeiro modelo utilizado neste trabalho e todos os demais foram aeronaves de propulsão elétrica, ou seja, usaram energia de uma ou mais baterias como “combustível” da motorização.

3.1.2 Asa Rotativa

Após a revolução comunista de 1917, o russo Igor Sykorsky migrou da antiga União Soviética para os Estados Unidos da América e começou seus trabalhos de pesquisa sobre aeronaves de asa rotativa (PEREIRA, 2010). Seu objetivo foi colocar em movimento circular as asas estacionárias de um avião, desenvolvendo, então, um novo aparelho para a aviação.

Sykorsky foi o pioneiro nesse ramo da aviação e, em meados da década de 1940, já produzia novos desses aparelhos em série.

Dentro de asas rotativas, pode-se citar duas importantes subdivisões: helicópteros e quadricópteros. Os primeiros são basicamente modelos dos helicópteros que tradicionalmente se veem nos céus, com um rotor principal e um rotor de calda. Já os quadricópteros utilizam quatro rotores iguais.

Os helicópteros geram o seu próprio fluxo de ar por meio de seu rotor principal e, por isso, o ambiente a pairar se torna bastante turbulento. Essas correntes, fluindo através das superfícies de controle e fuselagem do aparelho, geram ao piloto uma constante necessidade de correção e reposicionamento manual, o que pode acarretar atrasos, falha na missão ou, até mesmo, um acidente (PEREIRA, 2010). Além dessa característica, helicópteros, no geral, exigem um controle muito maior para sua estabilização do que aeronaves de asa fixa.

A Figura 18 mostra um helicóptero de resgate em ação perto da água. Nesta imagem, é possível notar claramente a ação das correntes de ar geradas pelo rotor principal.



Figura 18 - Ação das correntes de ar gerada pelo helicóptero

Fonte: (AVIATION, 2016)

Da mesma forma que as aeronaves de asa fixa, dois helicópteros foram utilizados no projeto. O primeiro modelo utilizado foi mais simples e da classe 450; o segundo, mais robusto e da classe 700. Este possuía capacidade de carga maior. Tais helicópteros são mostrados nas Figuras 19 e 20, respectivamente.



Figura 19 - Helicóptero utilizado (Classe 450)
Fonte: autoria própria.



Figura 20 - Helicóptero utilizado (Classe 700)
Fonte: autoria própria.

Os nomes dos modelos adquiridos são: o E-Sky Belt-CP V2, mostrado na Figura 19, como o primeiro helicóptero de classe 450, e o T-Rex 700E, Figura 20, citado como o segundo helicóptero utilizado no trabalho da classe 700.

Para a escolha desses modelos, foram especificados os seguintes requisitos:

- modelos do tipo acrobático, pois devem possuir uma boa relação peso-potência, para possibilitar a instalação de sistemas eletrônicos sem que essa carga extra tenha grande impacto no voo dos helicópteros;
- possuir o sistema de rotor principal do tipo Bell-Hiller, para suavizar os esforços exigidos sobre os servos-atuadores de controles cíclico e coletivo, pois este modelo de rotor possibilita comandos suaves e de rápida resposta;
- possuir um rotor de cauda do tipo *belt-cp*, ao contrário de muitos outros que possuem um motor elétrico separado. O fato de o rotor de cauda ser diretamente acionado pelo motor principal, ajustando-se apenas o *pitch* das pás, proporciona uma resposta de comando muito eficiente. Esse modelo de rotor de cauda é o mesmo modelo empregado em aeronaves reais (PEREIRA, 2010);
- possuir características que se aproximem ao máximo possível dos modelos de helicópteros reais, para que, em um trabalho futuro, possa haver a expansão mais fácil do trabalho; e
- é necessário ser um modelo já difundido no mercado, que possibilite uma busca mais fácil por peças de reposição, em caso de necessidade por motivo de queda ou acidente.

Por último, foi utilizado um quadricóptero, conforme ilustrado na Figura 21. Nota-se a montagem do GPS na parte superior da tampa do quadricóptero e os quatro motores e hélices de propulsão.



Figura 21 - Quadricóptero utilizado
Fonte: autoria própria.

3.2 Montagem e Configuração do *Hardware* e do *Software*

Para a montagem do *hardware* usado neste trabalho, fazem-se referências aos dispositivos enumerados no Capítulo 2. Primeiramente, foi necessário soldar a IMU ao fundo da placa de autopiloto Lisa/M. Os dispositivos restantes são todos ligados à placa de piloto automático mediante conectores *molex picoblade*, exceto os servos da aeronave e a placa PPM *encoder*, ligados por meio de conectores de servos comumente empregados no aeromodelismo tradicional.

Na saída, denominada UART3, da placa LISA/M, será conectada o módulo de GPS. O *modem* de comunicação será ligado a UART2 e o servo 6 da Lisa/M será conectado à PPM *encoder*, que, por sua vez, terá conectada todos os sinais do receptor de rádio controle, completando-se, assim, todos os dispositivos embarcados nas aeronaves. A Figura 22 mostra os dispositivos conectados, conforme mencionado, e uma caneta comum, que serve apenas como referência do tamanho dos dispositivos.

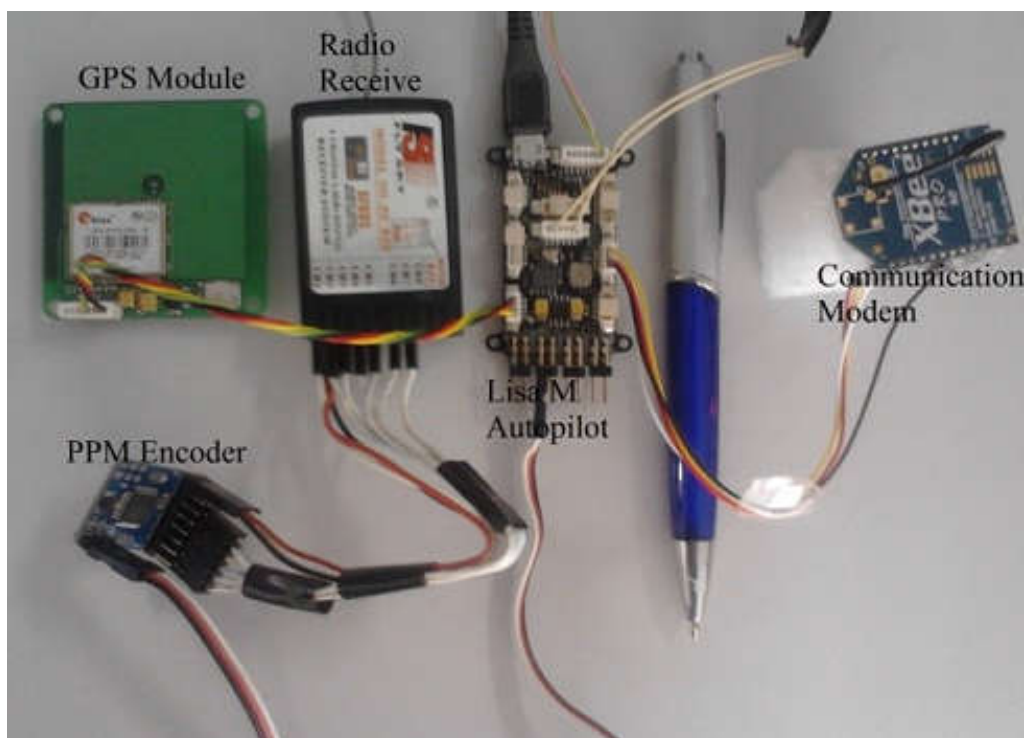


Figura 22 - Montagem dos dispositivos que irão embarcados

Fonte: autoria própria.

Outro dispositivo importante no envio e na recepção de mensagens entre o GCS e a aeronave é o outro *modem*. Este foi ligado ao *notebook*. Entretanto, para isso, foi necessária

uma placa intermediária chamada de FTDI. Ela conecta ao *modem* e tem uma saída USB para ser ligada no dispositivo com facilidade. A Figura 23 mostra o *modem* e sua placa FTDI, que irão ser conectadas ao *notebook*.



Figura 23 - Placa FTDI e modem de comunicação
Fonte: autoria própria.

É necessário ainda configurar o *software*. Para isso, iniciou-se a instalação da estação de terra, utilizando-se o sistema operacional Linux Ubuntu, versão 12.04. O código fonte do paparazzi está hospedado no Github. Para realizar a “clonagem” do repositório com o *git*, o seguinte comando foi utilizado:

```
git clone --origin upstream https://github.com/paparazzi/paparazzi.git
```

Nesse momento, outro comando se fez necessário:

```
git checkout v5.6
```

Para realizar a compilação do código fonte, foi necessário entrar no diretório do paparazzi, geralmente com o comando *cd ~/paparazzi*, e, então, executar o comando *run*. Por fim, para iniciar o *software*, usou-se o comando *./paparazzi*.

A partir desse momento, já foi possível se executar uma simulação na Central Paparazzi, escolhendo-se Microjet dentro de “*aircraft*”, selecionando-se “*sim*” como *target*. Depois, clicando-se em “*Build*”, esperando-se alguns segundos até que fosse finalizado o processo de construção da simulação e, logo após, clicando-se em “*Execute*” com a sessão de “*Simulation*” já selecionada.

Para executar o código em uma das aeronaves, tomou-se como padrão inicial o modelo *Bixler* encontrado nos arquivos do sistema paparazzi. Para executar o código e inseri-lo dentro da placa Lisa/M, foi necessário alterar o “*Target*” dentro do Paparazzi Center, para “*ap*”, clicar em “*Build*” e, depois realizar o “*Upload*”, de modo que a placa recebesse os arquivos configurados para a aeronave apropriada.

As principais alterações feitas podem ser encontradas nas primeiras linhas do arquivo da aeronave. Elas servem para selecionar se o modelo utilizado será uma aeronave de asa fixa ou uma aeronave de asas rotativas e, também, para identificar qual placa de piloto automático será utilizada. Como um exemplo, o código ficaria da seguinte forma:

```
<firmware name="fixedwing or rotorcraft"> <!-- Selecione o tipo de aeronava -->
  <target name="ap" board="lisa_m_2.0"/> <!-- Selecione a placa utilizada -->
  ...
</firmware>
```

Outro detalhe importante, ao se usarem transmissores com sinal PPM, como no caso deste trabalho de pesquisa, o arquivo de rádio deve ser configurado com extremo cuidado. Nesse arquivo, é possível encontrar todos os canais do rádio com suas respectivas funções, inclusive a função de seleção de modo, na qual é possível trocar entre comando manual, comando manual com estabilização automática e comando completamente autônomo.

3.3 Vídeo e Gravação de Imagens em VANTs

As principais aplicações para veículos aéreos não tripulados envolvem a captação de imagens, seja em formato de fotos ou vídeos para análise instantânea ou posterior. Tal análise poderá ser feita por um ser humano ou também, de forma automatizada, por *softwares* de tratamento de imagens e vídeos.

Neste trabalho, foi feita a análise das captações de fotos e também de vídeo. As duas formas de captação e exibição de vídeo são analógicas e digitais, sendo que a primeira forma tem custo reduzido e a segunda é mais adequada para imagens de alta definição (HABER; NOSSEN, 1977).

Quanto às câmeras, têm-se algumas opções: CCD (Dispositivo de Carga Acoplada) e CMOS (Semicondutor de Metal Óxido Complementar). Sensores de imagem CCD e CMOS são duas tecnologias diferentes para a captura de imagens, sendo que cada um tem seus pontos fortes e fracos, que dão vantagens em diferentes aplicações (CARLSON, 2002). Câmeras CCD tendem a ter melhor qualidade e menos ruído do que câmeras CMOS da mesma faixa de preço. Porém, as CMOS utilizam menos energia e são mais utilizadas em ambientes que utilizam 5V DC na alimentação.

Além das câmeras e dispositivos de captura e transmissão de imagens propriamente ditas, é importante que um sistema de uma aeronave não tripulada tenha também um equipamento para dar liberdade à câmera, fazendo com que ela não seja fixa em um ponto e possa ter movimentos independentes dos da aeronave. Esses mecanismos são chamados de *Pan-Tilt* ou *Gimbals*, em que, com alguma programação específica no *firmware* do sistema, tornam-se capazes de se movimentar, tentando seguir algum objeto ou um ponto específico, ou mesmo contribuir na estabilização das imagens (HONGKAI; XIAO GUANG; MIN, 2014).

A cada dia que passa, mais VANTs vêm sendo utilizados para captação de imagens aéreas com diversos objetivos, desde vigilância até o puro entretenimento. A Figura 24 mostra um dos drones sendo utilizado para realizar fotografias aéreas da Universidade Federal de Itajubá. Contudo, devido ao seu tamanho reduzido e à estrutura limitada, é muito difícil evitar vibrações mecânicas causadas pelo motor e pelo vento; conseqüentemente, as imagens enviadas podem não estar muito nítidas devido à instabilidade aplicada aos sensores da câmera, principalmente na transmissão de vídeo.



Figura 24 - Aplicação de Vant em fotografia aérea
Fonte: autoria própria.

Técnicas de estabilização de vídeo têm sido estudadas por pesquisadores, durante décadas, e muitos esquemas de estabilização têm sido propostos. Esses métodos podem ser classificados em duas categorias (SHEN *et al.*, 2009), ou seja, estabilização de vídeo com base mecânica e estabilização de vídeo baseada em processamento de imagens.

O primeiro método de estabilização utiliza *hardware* e dispositivos mecânicos, como, por exemplo, o *gimbal*, para reduzir a vibração. Já a estabilização baseada em processamento de imagens usa técnicas e algoritmos desenvolvidos para PDI (Processamento de Imagens), de forma a tornar a imagem mais nítida ao seu propósito (WANG, YUE *et al.*, 2011). Neste trabalho, ambas as técnicas foram utilizadas com o sistema Papparazzi.

3.4 Configuração de vídeo utilizada

Nesta seção, serão detalhadas as configurações de *hardware* e de *software* utilizadas nas aplicações deste trabalho, assim como a configuração necessária para o seu correto funcionamento.

3.4.1 *Hardware* do Sistema de Vídeo

Iniciando com o *hardware*, descrever-se-ão as câmeras analógica e digital. Conforme ilustram as Figuras 25 e 26, utilizando-se, principalmente, duas câmeras neste trabalho: a primeira, uma GoPro Hero3, capaz de filmar em alta resolução (HD) e gravar tudo em um cartão de memória para posterior análise ou mapeamento. Uma das utilidades dessa câmera foi tirar fotos sequenciais de uma região e, posteriormente, tratá-las formando um mapa da região em questão. A segunda, uma câmera FPV (Visão de Primeira Pessoa) CCD, utilizada na captação e transmissão em tempo real de imagens.



Figura 25 – Câmera GoPro Hero3
Fonte: autoria própria.



Figura 26 - Câmera FPV CCD
Fonte: autoria própria

Além das câmeras, fazem-se necessários também um receptor e um transmissor de vídeo, que, no caso, foi empregado em um par de aparelhos com as seguintes especificações: um transmissor analógico com 8 canais, 1.2 Ghz de frequência e 800mW de potência e o receptor com 12 canais. Além disso, as especificações do aparelho listam que ele pode ter o alcance de mais de 1 km com as antenas originais e as antenas podem ser substituídas por antenas com recepção mais sensíveis, o que foi aplicado neste trabalho. A Figura 27 mostra os aparelhos e as antenas de recepção originais.



Figura 27 - Vídeo Rx, Tx
Fonte: autoria própria

A antena omnidirecional de 3 dbi do receptor, vide Figura 27, foi trocada por uma antena *patch* de 8 dbi, acoplada a um tripé, conforme mostrado na Figura 28, ressaltando-se que ambas utilizam a conexão conhecida como *pigtail* para se ligar ao receptor.



Figura 28 - Antena patch de 8dbi
Fonte: autoria própria.

Para que as imagens recebidas no receptor de vídeo sejam exibidas na *ground station* (neste caso, em um *notebook*), foi necessário um equipamento de conversão de vídeo, o qual conecta o receptor de vídeo a uma porta USB, no caso, a chamada placa de captura USB da EasyCap, conforme ilustrado na Figura 29. Ela liga a saída de vídeo RCA no receptor de vídeo e se conecta no computador por meio de uma porta USB.



Figura 29 - Placa de Captura USB
Fonte: autoria própria.

Por fim, tem-se o conjunto *pan-tilt* ligado a dois servos comuns usados em aeromodelismo, conforme mostrado na Figura 30.



Figura 30 - Sistema Pan-Tilt com dois Servos
Fonte: autoria própria

3.4.2 *Software* do Sistema de Vídeo

Além da aquisição e montagem dos aparelhos e dispositivos de *hardware*, alguma configuração em *software* se faz necessária para o correto funcionamento e exibição das imagens enviadas em tempo real para o *notebook*.

Como o sistema operacional usado foi o Linux, as instruções mostradas na Figura 31, também foram necessárias para o referido sistema operacional.

```
sudo apt-get install mplayer (instalar codecs e player)

mplayer tv:// -tv driver=v4l2:width=640:height=480:device=/dev/video1
//Video 0 for webcam (usually)
//Video 1 for easycap (0 if no webcam)
```

Figura 31 - Comandos para instalação do reprodutor de mídia

Fonte: autoria própria.

A primeira linha de código da Figura 31 exibe o comando responsável pela instalação do reprodutor de mídia utilizado para exibição das imagens. Esse comando instala os *codecs*, o *player* e suas dependências para o correto funcionamento. A segunda linha executa o programa com os parâmetros enviados, inclusive os parâmetros de largura e altura, além de informar qual saída de vídeo deverá ser exibida. Na maioria dos *notebooks*, a saída “video0” é utilizada para a *webcam* do *notebook*. Portanto, a saída “video1” foi utilizada, neste trabalho, e cada usuário deve verificar essa peculiaridade.

As operações descritas na Figura 31 apenas possibilitam abrir o programa em outra janela, na qual será exibido o vídeo. É possível fazer com que o vídeo seja exibido dentro da estação de controle em terra do paparazzi. As instruções das Figuras 32 e 33 têm esse objetivo.

```
Alterar a linha do arquivo: ../conf/control_panel.xml

<program name="GCS" command="sw/ground_segment/cockpit/gcs -layout
horizontal.xml -mplayer 'tv:// -tv
driver=v4l2:width=640:height=480:device=/dev/video0'">
  <arg flag="-b" variable="ivy_bus"/>
</program>
```

Figura 32 - Comandos para adicionar o vídeo à GCS

Fonte: autoria própria.

A primeira linha da Figura 32 apenas indica o arquivo que deve ser alterado, `control_panel.xml`, que fica dentro da pasta “conf”, na qual foi instalado o `paparazzi`. Em seguida, deve-se encontrar a primeira linha dentro do arquivo e adicionar as linhas que estão faltando para que o `player` seja executado dentro do GCS. Mesmo com isso feito, o `software` ainda não saberá como posicionar exatamente o `player` e outro arquivo deve ser alterado. O seu nome é “horizontal.xml” e encontra-se localizado dentro das pastas “conf” e “gcs”, em que está instalado o programa. A Figura 33 exemplifica como deve ficar parte do arquivo após as alterações necessárias.

Alterar a linha do arquivo: ../conf/gcs/horizontal.xml

```
<layout width="1024" height="768">
<rows>
<columns>
  <widget size="500" name="map2d"/>
  <widget size="500" name="plugin"/>
</columns>
<columns>
  <rows size="375">
  ...
```

Figura 33 - Comandos para posicionar o vídeo no GCS

Fonte: autoria própria.

O próximo capítulo irá cobrir aplicações desenvolvidas e utilizadas durante esta pesquisa com os VANTs. A primeira propicia a análise de imagem térmica e a segunda trata do processamento de imagens aéreas.

4 Aplicações de VANTs autônomos de baixo custo

A utilização de imagens aéreas de alta resolução provenientes de Veículos Aéreos Não Tripulados (VANTs) para a extração de informações métricas da superfície terrestre tem sido vastamente pesquisada nos últimos anos. As facilidades existentes na realização de voos autônomos, associados a baixo custo, são os principais atrativos dessa nova tecnologia em aplicações para cobertura aerofotográfica (MITISHITA *et al.*, 2014).

Devido às facilidades existentes na realização do recobrimento aerofotográfico, as aplicações das tecnologias utilizadas em VANTs são mais apropriadas para a execução de projetos de engenharia que visam à extração de informações atualizadas da superfície terrestre (MITISHITA *et al.*, 2014). Este capítulo trata justamente de algumas aplicações desenvolvidas durante esta pesquisa.

4.1 Rede Neural Artificial e Análise de Imagem Térmica

Uma das aplicações dos veículos aéreos não tripulados é a inspeção de linhas de transmissão e transformadores de energia. Para realizar tais operações, é necessário que o VANT sobrevoe áreas próximas a torres de transmissão, portando uma câmera com sensor de infravermelho, para captar o calor emitido pelos equipamentos onde passa a energia elétrica e analisar essas imagens procurando por possíveis problemas.

Para fazer tais análises, foram utilizadas técnicas de Rede Neural Artificial (RNA), também conhecidas como ANN (*Artificial Neural Network*). RNA é equivalente a uma representação matemática do processamento biológico da informação transportada através do cérebro. Basicamente, uma RNA é composta por um conjunto de neurônios, cada um interligado a vários outros (FELIZARDO, LUIZ FLÁVIO *et al.*, 2015). A Figura 34 mostra uma rede neural com a sua arquitetura em “A” e um exemplo de neurônio em “B”.

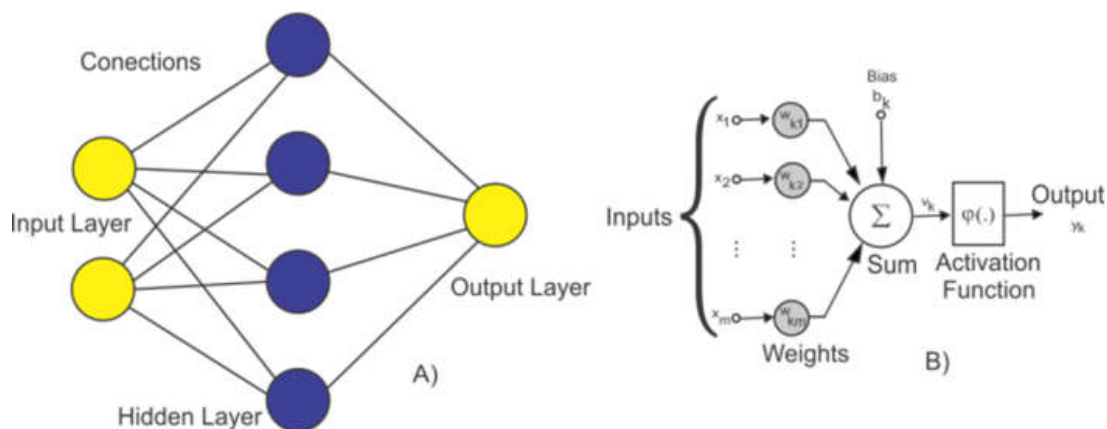


Figura 34 - Exemplo de RNA: A) Arquitetura e B) Neurônio

Fonte: autoria própria.

A informação é transmitida a partir da entrada para a saída, com os seus respectivos pesos (valores numéricos). Uma rede neural funciona em duas fases:

1. Aprendizagem, na qual um algoritmo apropriado processa cada nó e cada *link* entre eles, ajustando os seus pesos, a fim de encontrar o valor de saída desejado; e
2. Reconhecimento de padrões, quando, após a fase de aprendizado, a RNA identifica a informação de entrada, processa e emite o resultado desejado na saída, funcionando como um sistema de reconhecimento de padrões.

Observando a parte “B” da imagem, pode-se observar um neurônio, também conhecido como *perceptron*. Ele é considerado como a unidade básica de processamento de uma RNA (GRAUPE, 1997). Quando mais de uma camada intermediária de *perceptrons* é utilizada, a rede é chamada de MLP (*Multi Layer Perceptron*).

É admissível “ensinar” um padrão para a MLP, mediante um conjunto de entradas e respostas desejadas, durante a fase de treinamento. Analisando-se essas entradas e saídas da rede neural, é possível reconhecer o modelo (ou padrão) que está por trás. Durante o treino, a saída é comparada com a resposta esperada e os pesos sinápticos são ajustados até que o erro seja minimizado.

Foi feita uma pesquisa bem-sucedida utilizando-se a temperatura como um ponto de partida. Primeiramente, foi necessário determinar quais características definem o correto funcionamento do equipamento e, em seguida, extrair essas características da imagem. Como exemplo, pode-se imaginar que a temperatura de cada parte de um equipamento genérico tem o seu próprio intervalo ideal de operação. Em outras palavras, há um padrão de temperatura que caracteriza uma operação suave.

A Figura 34 mostra um exemplo de fotografia térmica de um transformador, no qual várias temperaturas são medidas, sendo que cada uma é representada por uma cor diferente.

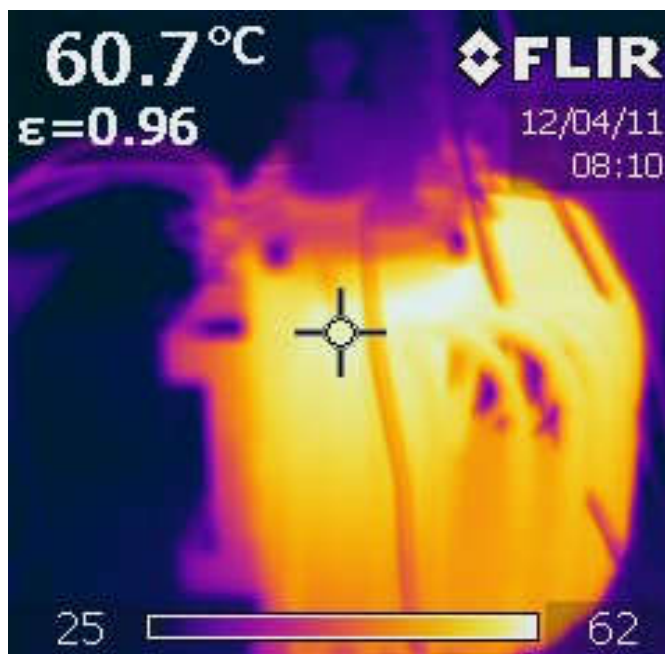


Figura 35 - Imagem térmica de um transformador
Fonte: autoria própria.

Para se testar a rede, criou-se um padrão aleatório e treinou-se a rede para ver se esta poderia fazer a identificação. As entradas são cinco valores de temperatura referentes a cada uma das partes de tal equipamento. Foi gerado, aleatoriamente, um conjunto de 150 entradas e as saídas desejadas foram definidas. A saída desejada foi definida como “1” se todas as temperaturas estiverem dentro de sua respectiva faixa de operação. Caso contrário, a saída desejada foi definida como “0”. A Tabela 2 representa as faixas de operação ótimas para a temperatura de cada parte do equipamento (FELIZARDO, LUIZ FLÁVIO *et al.*, 2015).

Tabela 2 - Padrões de temperatura projetados para testar a eficiência da MLP

Partes	Faixa de Operação Desejada °C
parte 1	[20,30]
parte 2	[35, 50]
parte 3	[80, 100]
parte 4	[150, 200]
parte 5	[5, 10]

Fonte: autoria própria.

Utilizando a ferramenta de rede neural do MATLAB (versão R2008a), as configurações da rede foram feitas da seguinte forma:

- Número de Neurônios = 50;
- Número de Camadas = 2;
- Função de Transferência = Sigmoide;
- Algoritmo de Treinamento = *trainscg backpropagation*; e
- 70% das entradas foram usadas para o treinamento, 15% para validação e 15% para testes.

Depois da rede treinada, uma simulação foi feita para verificar se ela havia “aprendido” o padrão desejado. O resultado (ans) obtidos para as três diferentes simulações foram:

Simulação 1:

```
>> sim (net, [25; 40; 90; 170; 8])
ans = 0.9981 (error = 0,19%)
```

Simulação 2:

```
>> sim (net, [10; 40; 90; 170; 8])
ans = 0.1303 (error = 13,03%)
```

Simulação 3:

```
>> sim (net, [10; 40; 90; 170; 12])
```

```
ans = 0.0199 (error = 1,99%)
```

Analisando-se os resultados, provou-se que, de fato, a rede conseguiu fazer a generalização e identificar o padrão. Na prática, outras características, além de temperatura, podem ser usadas como entrada. Apenas, é necessário se considerar, no equipamento em uso, quais os pontos que caracterizam o seu funcionamento perfeito ou não.

4.2 Mapas Kohonen e processamento de imagens aéreas

Atualmente, é muito comum encontrar áreas que precisam ser monitoradas, de modo a garantir a sua integridade, seja evitando-se invasões, avaliando-se desmatamentos, verificando-se alterações não autorizadas em terreno e, até mesmo, no caso de ocupação, em terrenos que têm pagamento de impostos. O uso de UAVs para monitoramento aumentou consideravelmente nos últimos anos, especialmente com o uso combinado de dados de sensores, como o Sistema de Posicionamento Global (GPS) e a Unidade de Medição Inercial (IMU), para realizar o registro da localização das imagens (SEMSCH *et al.*, 2009) (WANG, YI; SCHULTZ; FEVIG, 2009).

Para esse tipo de classificação em imagens, é necessário o uso de técnicas de inteligência artificial. Neste trabalho, foi usada uma específica, chamada de SOM Kohonen (Mapa Kohonen Auto-organizável).

Um sistema de processamento de imagens tem geralmente as seguintes partes (GONZALEZ; WOODS, 2006):

- Aquisição de imagens – consiste de um sensor para captar a imagem e um segundo elemento capaz de digitalizar o sinal, gerando uma imagem digital;
- Pré-processamento – tem a função de melhorar a imagem, buscando aumentar as chances de sucesso para os procedimentos seguintes. Envolve técnicas de realce de contraste, remoção de ruído e isolamento de regiões;
- Segmentação – divide uma imagem em partes ou objetos. É um dos processos mais difíceis no processamento de imagem. A saída deste estágio é geralmente dada sob a forma de pixels e corresponde ao limite de uma região;
- Representação e descrição – nesta fase do processamento, os dados são transformados de forma adequada para o processamento computacional que se segue, destacando características de interesse. O processo de descrição, também chamado de seleção de características, extrai características que representam a informação quantitativa de interesse ou aquelas fundamentais na descrição de classes e objetos; e
- Reconhecimento – este é o processo que atribui uma etiqueta a um objeto e envolve a atribuição de significado a um conjunto de objetos reconhecidos.

A técnica de SOM Kohonen envolve um mapa ou matriz com dimensão não simétrica, em que cada elemento representa um neurônio (MACHADO; MACHADO, 2013). Os neurônios são interligados uns aos outros e até mesmo se influenciando. Cada neurônio ou grupo de neurônios representa uma saída e é responsável por uma função particular. Essa

característica é análoga ao cérebro, no qual informações diferentes são controladas por diversas partes do cérebro, como fala, audição e visão por exemplo.

A rede neural artificial se assemelha a outros aspectos do cérebro, tais como: aquisição de conhecimento do ambiente por meio de um processo de aprendizagem e de conexão fortes entre os neurônios – os pesos sinápticos –, que são usados para armazenar o conhecimento adquirido.

Os pesos sinápticos são inicializados aleatoriamente. Durante o aprendizado, esses pesos são atualizados a cada iteração do algoritmo. A intenção dessa fase é encontrar o neurônio que tem as características mais próximas da entrada. Assim, após um processo de iterações, apenas um neurônio é ativado. No entanto, os seus vizinhos sofrem uma pequena influência.

Nesta pesquisa, foi desenvolvido um programa em Java, que implementa a técnica do SOM Kohonen. Esse programa permite a entrada de uma imagem e gera uma nova imagem de saída, na qual os pixels são agrupados de acordo com características semelhantes.

Para o processamento, alguns parâmetros devem ser definidos. Esses parâmetros estão ligados ao tipo de sensor, sendo utilizado para capturar as imagens. Por exemplo, os parâmetros ótimos para imagens capturadas por um VANT não são os mesmos para imagens de satélite. De acordo com Mota *et al.* (2013), estes parâmetros são:

- Número de iterações – quantas vezes o algoritmo será repetido. Teoricamente, quanto maior for o número de iterações, maior é a qualidade do tratamento. Entretanto, não se pode aplicar demasiadamente esse processo repetitivo, visto que o número excessivo de iterações começará a prejudicar a qualidade da imagem de saída. Assim, um número intermediário deve ser selecionado, de modo que a qualidade não seja comprometida;

- Taxa de aprendizagem – a taxa de aprendizagem interfere com a atualização dos pesos sinápticos. Taxas de aprendizagem muito pequenas requerem maior número de iterações para convergirem em um resultado desejado. No entanto, uma taxa muito elevada provoca oscilações e também dificulta a convergência; e

- Número de neurônios – cada neurônio representa uma classe, isto é, uma característica distinta. O número de neurônios deve ser colocado de acordo com a forma como o usuário quer ter alguma generalização no agrupamento de neurônios. Quanto menor o número de neurônios, mais generalizado o agrupamento será. Um número muito elevado de neurônios tenderá a fazer com que a saída seja uma cópia da entrada, o que não é desejado.

Para testar a aplicação da técnica SOM Kohonen e compreender melhor o seu funcionamento, alguns testes foram conduzidos, usando imagens aéreas de uma cidade. A Tabela 3 mostra os parâmetros usados na configuração (FELIZARDO, LUIZ F. *et al.*, 2013):

Tabela 3 - Atributos utilizados nos testes feitos com o SOM Kohonen

Test #	Número de iterações	Taxa de Aprendizagem	Número de Neurônios
Teste 0	3	0.000001	2x1
Teste 1	4	0.000001	2x1
Teste 2	5	0.000001	2x1
Teste 3	6	0.000001	2x1
Teste 4	7	0.000001	2x1

Fonte: autoria própria.

A sequência de passos tomados na implementação dos testes foram:

1. As imagens originais foram convertidas em imagens com tons de cinza;
2. Foi feito um ajustamento no brilho e no tom de cor das imagens; e
3. A imagem foi, então, processada pelo *software* que se utiliza da técnica SOM Kohonen.

As Figuras 36, 37, 38 e 39 mostram quatro imagens em que serão demonstrados os resultados do Teste 2 da Tabela 3.



Figura 36 - Teste SOM, imagem original de entrada

Fonte: autoria própria.

Na Figura 36, pode-se observar a fotografia original retirada sobre a cidade de São José dos Campos – SP.



Figura 37 - Teste SOM, imagem após a aplicação do passo um
Fonte: autoria própria.

A Figura 37 mostra o resultado após a imagem ter passado pelo passo 1 da sequência mencionada anteriormente.



Figura 38 - Teste SOM, imagem após o passo dois
Fonte: autoria própria.

Na Figura 38, a imagem já teve seu brilho e tons alterados e está pronta para o último passo do estudo de caso.

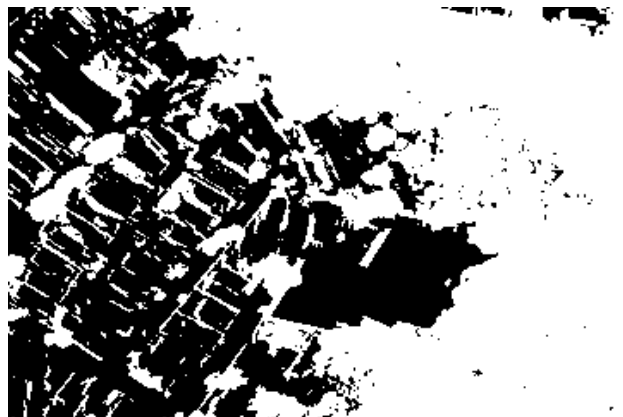


Figura 39 - Teste SOM, imagem de saída

Fonte: autoria própria.

Finalmente, a Figura 39 exhibe o resultado final após todos os passos.

Os testes foram feitos usando apenas dois neurônios, um representando a cor branca e outro representando a cor preta. Os passos 1 e 2 correspondem a um pré-processamento que auxilia o processamento final da imagem realizado, utilizando-se a técnica SOM Kohonen na etapa 3. O pré-processamento torna mais fácil para o processo final reconhecer qual cor está mais próxima da branca e qual está mais perto da cor preta.

O resultado final exibido na Figura 39 pode ser comparado com outro resultado final de uma imagem tirada no ano seguinte por exemplo. Caso alguma parte em preto apareça nessa nova imagem significa que aquele terreno antes coberto por árvores foi invadido, a árvore cortada e uma punição poderá ser aplicada ao responsável.

5 Conclusão

O uso de Veículos Aéreos não Tripulados para os mais diversos objetivos tem aumentado consideravelmente nos últimos anos. Com este trabalho finalizado, pode-se concluir que é possível montar e configurar aeromodelos tornando-os autônomos e com um baixo custo utilizando-se de tecnologias *Open Source*.

Além de realizar a integração de *software* e *hardware* com as configurações e montagem dos sistemas aéreos, também foi possível, com a utilização de técnicas de inteligência artificial, tratar imagens, a fim de colocar um uso prático para as aeronaves utilizadas. Dois casos de usos diferentes foram tratados no trabalho: o primeiro tendo como resultado encontrar falhas em dispositivos de transmissão de energia e o segundo caso buscando analisar imagens de uma mesma área em diferentes períodos para detecção de desmatamentos ilegais.

Com alguns ajustes, o sistema pode ser usado para outros fins. Alguns exemplos de recomendações são: inspeção de áreas rurais em busca de plantações ilegais ou pistas clandestinas, filmagem e fotografias aéreas em geral como procura de possíveis focos da dengue e apoio em operações militares como invasões em território inimigo para pré-checagem de posicionamento.

Como sugestão de trabalhos futuros pode-se citar a implementação de um sistema autônomo usando celulares e *tablets* como estações centrais de controle. Outros exemplos seriam a melhoria ou troca dos sistemas de estabilização PID por outros tipos de sistemas, a melhoria do sistema de GPS para obter uma melhor precisão ou o desenvolvimento de um sistema de reconhecimento de obstáculos.

Referências

AMAHAH, J. The Design of an Unmanned Aerial Vehicle Based on the ArduPilot. *Georgian Electronic Scientific Journal: Computer Science and Telecommunications*, v. 5, n. 5, p. 144–153, 2009.

ARDUPILOT. *ArduPilot Autopilot Suite*. Disponível em: <<http://plane.ardupilot.org/ardupilot/index.html>>. Acesso em: 15 abr. 2016.

AUSTIN, R. *Unmanned Aircraft Systems: UAVS design, development and deployment*. [S.l.]: John Wiley & Sons, 2010.

AVIATION, S. *Aviation Spectator*. Disponível em: <<http://www.aviationspectator.com/files/images>>. Acesso em: 10 abr. 2016.

BEHNCK, L. P. *Controle de Missão de Voo de Veículo Aéreo Não-Tripulado Veículo Aéreo Não-Tripulado*. 2014. 59 f. Universidade Federal do Rio Grande do Sul, 2014.

BIN, H.; JUSTICE, A. The Design of an Unmanned Aerial Vehicle Based on the ArduPilot. *Indian Journal of Science and Technology*, v. 2, n. 4, p. 12–15, 2009. Disponível em: <<http://www.indjst.org/index.php/indjst/article/view/29423>>.

CARLSON, B. S. Comparison of modern CCD and CMOS image sensor technologies and systems for low resolution imaging. 2002, [S.l.]: IEEE, 2002. p. 171–176.

COLLINS DICTIONARY. *Collins English Dictionary: Complete and Unabridged*. 12. ed. [S.l.]: HarperCollins, 2014.

COMANDO DA AERONÁUTICA. *Regras para a Operação do Aerodelismo no Brasil*. [S.l.]: PORTARIA DAC Nº 207/STE DE 07 DE ABRIL, 1999.

FELIZARDO, L. F. *et al.* Integration of ANN and UAV for Aerial Inspection. *International Micro Air Vehicles Conference and Flight Competition*, v. 1, 2015.

FELIZARDO, L. F. *et al.* Using ANN and UAV for terrain surveillance. 2013, [S.l.: s.n.], 2013. p. 1–5. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6920414>>.

GATI, B. Open source autopilot for academic research-The Paparazzi system. *American Control Conference (ACC), 2013*, p. 1478–1481, 2013. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6580045>.

GIL, A. C. *Como elaborar projetos de pesquisa*. [S.l.]: Atlas, 1946.

GONZALEZ, R.; WOODS, R. *Digital Image Processing*. 3ª. ed. [S.l.]: Prentice-Hall, 2006.

GRAUPE, D. *Principles of Artificial Neural Networks*. [S.l.]: World Scientific, 1997.

HABER, C.; NOSSEN, E. Analog Versus Digital Antijam Video Transmission. *IEEE Transactions on Communications*, v. 25, n. 3, p. 310–317, mar. 1977.

- HAYKIN. *Redes Neurais Principios e Prática*. 2^a. ed. [S.l.]: Porto Alegre, Bookman, 2001.
- HAYKIN, S. “*Self-organizing maps*”. *Neural networks - A comprehensive foundation*. 2nd. ed. [S.l.]: Prentice-Hall, 1999.
- HONGKAI, C.; XIAO GUANG, Z.; MIN, T. A novel pan-tilt camera control approach for visual tracking. jun. 2014, [S.l.]: IEEE, jun. 2014. p. 2860–2865.
- JACKSON, R. *Helicopter Power*. Disponível em: <<http://punkoryan.com/2009/10/06/helicopter-power>>. Acesso em: 29 maio 2015.
- KNUTH, S. L. *et al.* Unmanned aircraft system measurements of the atmospheric boundary layer over Terra Nova Bay, Antarctica. *Earth System Science Data*, v. 5, n. 1, p. 57–69, 2013. Disponível em: <<http://www.earth-syst-sci-data.net/5/57/2013/>>.
- KOHONEN. *Self-Organizing Maps*. 3rd. ed. [S.l.]: Springer, 2000.
- MACHADO, A.; MACHADO, L. *Neuroanatomia Funcional*. 3^a. ed. [S.l.]: Atheneu, 2013.
- MANSOUR, M. *et al.* Full Scale Wind Turbine Flowfield Measurements Using a 7-Sensor Fast Response Probe. 2010, [S.l.: s.n.], 2010. p. 871–882. Disponível em: <<http://proceedings.asmedigitalcollection.asme.org/proceeding.aspx?articleid=1609283>>.
- MITISHITA, E. *et al.* O Uso De Veículos Aéreos Não Tripulados (Vants) Em Aplicações De Mapeamento Aerofotogramétrico. 2014, Gramado: [s.n.], 2014. p. 10.
- MOTA, R. L. M. *Aplicação de Mapas Auto-Organizáveis na Detecção de Mudanças no Solo ao Longo do Tempo*. Itajubá, MG: [s.n.], 2014.
- PAPARAZZI. *Paparazzi UAV Wiki*. Disponível em: <<http://paparazzi.enac.fr/wiki/>>. Acesso em: 20 abr. 2016.
- PEREIRA, M. O. *Controle de estabilização para helicópteros não tripulados usando técnicas PID*. 2010. Universidade Federal de Lavras, 2010.
- SEMSCH, E. *et al.* Autonomous UAV Surveillance in Complex Urban Environments. 2009, [S.l.: s.n.], 2009. p. 82–85. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5284858>>.
- SHEN, Y. *et al.* Video stabilization using principal component analysis and scale invariant feature transform in particle filter framework. *IEEE Transactions on Consumer Electronics*, v. 55, n. 3, p. 1714–1721, ago. 2009.
- WANG, Y. *et al.* Real-Time Video Stabilization for Unmanned Aerial Vehicles. *Conference on Machine Vision Applications*, p. 336–339, 2011.
- WANG, Y.; SCHULTZ, R. R.; FEVIG, R. A. Sensor fusion method using GPS/IMU data for fast UAV surveillance video frame registration. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, p. 985–988, 2009.