



MINISTÉRIO DA  
CIÊNCIA, TECNOLOGIA  
E INOVAÇÕES



sid.inpe.br/mtc-m21c/2021/01.28.22.11-TDI

## NAVEGAÇÃO AUTÔNOMA DE VANT POR FUSÃO DE DADOS COM REDE NEURAL ARTIFICIAL OTIMIZADA IMPLEMENTADA EM FPGA

Gerson da Penha Neto

Tese de Doutorado do Curso de Pós-Graduação em Computação Aplicada, orientada pelos Drs. Haroldo Fraga de Campos Velho, e Elcio Hideiti Sheguemori, aprovada em 26 de janeiro de 2021.

URL do documento original:

<<http://urlib.net/8JMKD3MGP3W34R/4443NUP>>

INPE  
São José dos Campos  
2021

**PUBLICADO POR:**

Instituto Nacional de Pesquisas Espaciais - INPE  
Coordenação de Ensino, Pesquisa e Extensão (COEPE)  
Divisão de Biblioteca (DIBIB)  
CEP 12.227-010  
São José dos Campos - SP - Brasil  
Tel.:(012) 3208-6923/7348  
E-mail: pubtc@inpe.br

**CONSELHO DE EDITORAÇÃO E PRESERVAÇÃO DA PRODUÇÃO INTELLECTUAL DO INPE - CEPPII (PORTARIA Nº 176/2018/SEI-INPE):**

**Presidente:**

Dra. Marley Cavalcante de Lima Moscati - Coordenação-Geral de Ciências da Terra (CGCT)

**Membros:**

Dra. Ieda Del Arco Sanches - Conselho de Pós-Graduação (CPG)  
Dr. Evandro Marconi Rocco - Coordenação-Geral de Engenharia, Tecnologia e Ciência Espaciais (CGCE)  
Dr. Rafael Duarte Coelho dos Santos - Coordenação-Geral de Infraestrutura e Pesquisas Aplicadas (CGIP)  
Simone Angélica Del Ducca Barbedo - Divisão de Biblioteca (DIBIB)

**BIBLIOTECA DIGITAL:**

Dr. Gerald Jean Francis Banon  
Clayton Martins Pereira - Divisão de Biblioteca (DIBIB)

**REVISÃO E NORMALIZAÇÃO DOCUMENTÁRIA:**

Simone Angélica Del Ducca Barbedo - Divisão de Biblioteca (DIBIB)  
André Luis Dias Fernandes - Divisão de Biblioteca (DIBIB)

**EDITORAÇÃO ELETRÔNICA:**

Ivone Martins - Divisão de Biblioteca (DIBIB)  
André Luis Dias Fernandes - Divisão de Biblioteca (DIBIB)



MINISTÉRIO DA  
CIÊNCIA, TECNOLOGIA  
E INOVAÇÕES



sid.inpe.br/mtc-m21c/2021/01.28.22.11-TDI

## NAVEGAÇÃO AUTÔNOMA DE VANT POR FUSÃO DE DADOS COM REDE NEURAL ARTIFICIAL OTIMIZADA IMPLEMENTADA EM FPGA

Gerson da Penha Neto

Tese de Doutorado do Curso de Pós-Graduação em Computação Aplicada, orientada pelos Drs. Haroldo Fraga de Campos Velho, e Elcio Hideiti Sheguemori, aprovada em 26 de janeiro de 2021.

URL do documento original:

<<http://urlib.net/8JMKD3MGP3W34R/4443NUP>>

INPE  
São José dos Campos  
2021

Dados Internacionais de Catalogação na Publicação (CIP)

---

Penha Neto, Gerson da.

P376n Navegação autônoma de VANT por fusão de dados com rede neural artificial otimizada implementada em FPGA / Gerson da Penha Neto. – São José dos Campos : INPE, 2021.

xxvi + 185 p. ; (sid.inpe.br/mtc-m21c/2021/01.28.22.11-TDI)

Tese (Doutorado em Computação Aplicada) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2021.

Orientadores : Drs. Haroldo Fraga de Campos Velho, e Elcio Hideiti Sheguemori.

1. Fusão de dados. 2. Veículo aéreo não tripulado. 3. Rede neural autoconfigurada. 4. Field Programmable Gate Array.  
I.Título.

CDU 004.455.1:629.014.8

---



Esta obra foi licenciada sob uma Licença [Creative Commons Atribuição-NãoComercial 3.0 Não Adaptada](https://creativecommons.org/licenses/by-nc/3.0/).

This work is licensed under a [Creative Commons Attribution-NonCommercial 3.0 Unported License](https://creativecommons.org/licenses/by-nc/3.0/).





MINISTÉRIO DA  
CIÊNCIA, TECNOLOGIA  
E INOVAÇÕES



## INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

### DEFESA FINAL DE TESE DE GERSON DA PENHA NETO

No dia 26 de janeiro de 2021, às 09h, por videoconferência, o(a) aluno(a) mencionado(a) acima defendeu seu trabalho final (apresentação oral seguida de arguição) perante uma Banca Examinadora, cujos membros estão listados abaixo. O(A) aluno(a) foi APROVADO(A) pela Banca Examinadora por unanimidade, em cumprimento ao requisito exigido para obtenção do Título de Doutor em Computação Aplicada. O trabalho precisa da incorporação das correções sugeridas pela Banca Examinadora e revisão final pelo(s) orientador(es).

Novo Título: "NAVEGAÇÃO AUTÔNOMA DE VANT POR FUSÃO DE DADOS COM REDE NEURAL ARTIFICIAL OTIMIZADA IMPLEMENTADA EM FPGA"

Eu, Stephan Stephany, como presidente da banca, assino esta ATA em nome de todos os membros.

#### Membros da Banca

Dr. Stephan Stephany, Presidente INPE.  
Dr. Haroldo Fraga de Campos Velho, Orientador(a) INPE  
Dr. Elcio Hideiti Shiguemori, Orientador(a) INPE  
Dr. Lamartine Nogueira Frutuoso Guimarães, Membro da Banca, INPE  
Dr. Alexandre Carlos Brandão Ramos, Convidado(a) UNIFEI  
Dr. Antonio de Pádua Braga, Convidado(a) UFMG



Documento assinado eletronicamente por **Stephan Stephany, Pesquisador Titular**, em 28/01/2021, às 14:35 (horário oficial de Brasília), com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site <http://sei.mctic.gov.br/verifica.html>, informando o código verificador **6429827** e o código CRC **8467ECE2**.

---

**Referência:** Processo nº 01340.000557/2021-95

SEI nº 6429827

*“O universo é criação da Mente. E é na Mente que reside a intenção do Ato. Então, o Ato é materialização da Intenção”.*

FERNANDO SANCHEZ, 2020



*A **Luciana**, minha amada esposa, obrigado por todo  
apoio e motivação.*



## AGRADECIMENTOS

Em primeiro lugar, como sempre, agradeço a Deus, que é o grande arquiteto do universo. Muito obrigado Deus por permitir a realização desta etapa na minha vida. Entendo Deus como a energia que organiza tudo no universo. Sem Ele nada seria possível, nem mesmo o desenvolvimento da ciência.

Agradeço a toda minha família. Graças aos meus pais pude chegar neste nível de estudo e desenvolvimento pessoal. É importante agradecer a um membro em particular, meu tio Paulo Afonso. Ele é um grande professor para mim e um pai para muitas pessoas, um verdadeiro líder.

O agradecimento especial pertence a minha esposa, Luciana Cruz. Ela me apoiou e me ajudou em todo os momentos. Nunca me deixou desistir e sempre estava lá como conselheira nos momentos mais difíceis.

Por fim, agradeço aos meus orientadores, prof. Haroldo Fraga e prof. Elcio Shiguetori. Eles são grandes mentores e me ajudaram nesta caminhada.





## RESUMO

O uso de VANT aumentou nos últimos anos e devido o aumento do emprego de VANTs, cresceu também o interesse de empresas, organizações governamentais e da comunidade científica, no desenvolvimento de sistemas autônomos aéreos, principalmente de pequeno porte. A navegação necessita de sensores, como um Global Navigation Satellite System (GNSS) e um Inertial Navigation System (INS). Contudo, existem problemas associados ao uso destes sensores. Os principais problemas são interferências, que podem ser provocadas intencionalmente ou por ação natural, o que pode inviabilizar a navegação autônoma. Uma solução quando os sensores falham ou não estão disponíveis é a navegação por imagens. Contudo, ainda existem problemas associados a navegação autônoma, por processamento de imagens. Esses problemas advém da sua natureza técnica, que é sensível a interferências de fatores ambientais, como nuvens, chuva, fumaça, luminosidade, sensores imageadores de baixa qualidade ou qualquer outro fator que atrapalhe a coleta de imagens durante a navegação. Por isso, para mitigar possíveis falhas das alternativas, que utilizam apenas imagens, a solução sugerida na literatura é aplicar técnicas de fusão de dados. Com a fusão de dados, é possível obter maior confiabilidade a navegação autônoma de VANTs e desta forma, garantir uma melhor estimativa da posição do VANT durante a navegação. Uma das técnicas que se destaca, como fusor de dados, é o Filtro de Kalman (FK). Contudo, o FK possui desvantagens e dentre elas destaca-se a complexidade exigida na construção/implementação do FK. O objetivo desta Tese é investigar, analisar e qualificar a estimação da posição de um VANT, a partir da fusão dos dados, dos sensores embarcados, utilizando uma MLP autoconfigurada, como alternativa ao FK. Além de fornecer uma alternativa ao FK, também é um objetivo a construção de um dispositivo de processamento em alto desempenho, um hardware dedicado, para implementar a técnica investigada num dispositivo pequeno e possível de embarcar num VANT de pequeno porte.

Navegação autônoma. Fusão de dados. Veículo aéreo não tripulado. Rede neural autoconfigurada. Field Programmable Gate Array.



# AUTONOMOUS UAV NAVIGATION BY DATA FUSION WITH OPTIMIZED ARTIFICIAL NEURAL NETWORK IMPLEMENTED IN FPGA

## ABSTRACT

The use of UAVs has increased in recent years and due to the increase in the use of UAVs, the interest of companies, governmental organizations and the scientific community in the development of autonomous aerial systems, especially small ones, has also grown. Navigation requires sensors, such as a Global Navigation Satellite System (GNSS) and an Inertial Navigation System (INS). However, there are problems associated with the use of these sensors. The main problems are interferences, which can be caused intentionally or by action natural, which can make autonomous navigation unfeasible. One solution when sensors fail or are not available is image navigation. However, there are still problems associated with autonomous navigation, due to image processing. These problems are due to their technical nature, which is sensitive to interference from environmental factors, such as clouds, rain, smoke, light, low quality imaging sensors or any other factor that interferes with image collection during navigation. Therefore, to mitigate possible failures of the alternatives, which use only images, the solution suggested in the literature is to apply data fusion techniques. With the merger of data, it is possible to obtain greater reliability in the autonomous navigation of UAVs and in this way, guarantee a better estimate of the UAV position during navigation. One of the techniques that stands out, as a data fuser, is the Kalman Filter (FK). However, FK has disadvantages and among them stands out the complexity required in the construction / implementation of FK. The objective of this Thesis is to investigate, analyze and qualify the estimation of the position of a UAV, from the fusion of data, of the embedded sensors, using a self-configured MLP, as an alternative to the FK. In addition to providing an alternative to FK, it is also an objective to build a high performance processing device, dedicated hardware, to implement the investigated technique in a small device and possible to embark on a small UAV.

Keywords: Autonomous navigation. Data fusion. Unmanned aerial vehicle. Auto-configured neural network. Field Programmable Gate Array.



## LISTA DE FIGURAS

	<u>Pág.</u>
1.1	Três dimensões de movimento possíveis para um VANT. . . . . 4
1.2	Alcance de tecnologias com o padrão IEEE 802.11 (Wi-Fi). . . . . 5
1.3	Cintilação Ionosférica e Anomalia Magnética do Atlântico Sul. . . . . 6
1.4	Processo de navegação autônoma, por fusão de dados, para um VANT. . . . . 8
3.1	Elementos que compõe o GPS. . . . . 26
3.2	Abordagem utilizada para um DGPS. . . . . 27
3.3	Sistema inercial com acelerômetros e giroscópios. . . . . 28
3.4	Correlação cruzada entre a imagem do VANT e a imagem georeferenciada. 34
3.5	Aplicação do filtro Gaussiano. . . . . 38
3.6	Aspecto das derivadas da matriz intensidade. . . . . 41
3.7	Exemplos de padrões de bordas, para treinamento da MLP. . . . . 43
3.8	Representação de um padrão de bordas em vetor de pixels. . . . . 44
3.9	Deslocamento da janela deslizante para extração de bordas. . . . . 45
3.10	Analogia entre sensores e os órgãos do sentido. . . . . 46
3.11	Funcionamento da ativação do neurônio artificial. . . . . 52
3.12	Fluxo de construção/configuração do <i>hardware</i> em FPGA. . . . . 54
4.1	Metodologia, numa visão geral. . . . . 57
4.2	Transformação das imagens e primeira estimativa. . . . . 58
4.3	Fluxo de atividades para os experimentos da primeira fase. . . . . 61
5.1	VANT utilizado nos experimentos na Suécia. . . . . 63
5.2	Trajetória realizada pelo VANT nos experimentos na Suécia. . . . . 64
5.3	VANT utilizado no experimento no Brasil. . . . . 66
5.4	Trajetória realizada pelo VANT no experimento no Brasil. . . . . 67
6.1	Resultado da estimação sobre os conjuntos de validação e generalização, para o primeiro experimento. . . . . 73
6.2	Erro médio de generalização para cada $k$ conjunto. . . . . 78
6.3	Estimações de latitude para cada $k$ conjunto. . . . . 79
6.4	Estimações de longitude para cada $k$ conjunto. . . . . 80
6.5	Estimações de cada $k$ conjunto, seguindo o esquema de 120 combinações. 81
6.6	Erro modular horizontal. . . . . 86
6.7	Novo conjunto de dados, após a remoção de ruído. . . . . 87
6.8	Intervalos de remoção de ruído para latitude. . . . . 88
6.9	Intervalos de remoção de ruído para longitude. . . . . 89

6.10	Valores obtidos pelo giroscópio para cada eixo. . . . .	92
6.11	Suavização do filtro média móvel para $k = 100$ . . . . .	93
6.12	Seleção do tamanho da janela móvel, para o filtro média móvel. . . . .	94
6.13	Resultados da fusão de dados para os conjuntos com acelerômetro ou giroscópio. . . . .	102
7.1	Imagem georreferenciada, usada na estimação por visão computacional. . . . .	104
7.2	Linhas de voo e pontos de controle. . . . .	106
7.3	Trajectoria estimada por visão computacional. . . . .	108
7.4	Trajectoria estimada por visão computacional, após filtragem. . . . .	110
7.5	Padrões de bordas utilizados para treinamento do extrator de bordas neural. . . . .	111
7.6	Estimação de cada coordenada por visão computacional. . . . .	113
7.7	Imagens usadas para estimação da trajetória, utilizando a correlação cruzada. . . . .	114
7.8	Estimações obtidas pela fusão de dados com MLP autoconfigurada. . . . .	117
7.9	Estimações obtidas pela fusão de dados com MLP autoconfigurada. . . . .	120
7.10	Primeiros dispositivos usados para implementação e teste do fusor neural em FPGA. . . . .	126
7.11	Processo de envio e armazenamento de dados para dentro do fusor neural. . . . .	127
7.12	Implementação da função de ativação do neurônio com uma LUT. . . . .	128
7.13	Desenvolvimento da LUT usada na extração de bordas. . . . .	129
7.14	Trajectoria estimada pelo fusor neural em comparação com a trajetória medida pelo GPS-RTK. . . . .	130
7.15	Trajectoria estimada pelo fusor neural implementado no computador <i>Zybo Z7</i> . . . . .	134
7.16	Estimações obtidas pela fusão de dados com FPGA, no computador <i>Zybo Z7</i> . . . . .	136
7.17	Trajectoria estimada pelo fusor neural implementado no computador <i>Zybo Z7</i> , com ativação CORDIC. . . . .	137
7.18	Estimações obtidas pela fusão de dados com FPGA, no computador <i>Zybo Z7</i> , com ativação CORDIC. . . . .	140
7.19	Desenvolvimento da LUT com ativação interpolada. . . . .	141
7.20	Trajectoria estimada pelo fusor neural implementado no computador <i>Zybo Z7</i> e ativação LUT interpolada. . . . .	142
7.21	Estimações obtidas pela fusão de dados com FPGA, no computador <i>Zybo Z7</i> e ativação LUT interpolada. . . . .	144
7.22	Estimações para cada coordenada obtidas pelas ativações CORDIC e LUT interpolada. . . . .	145

8.1	Separação dos índices do desvio padrão por aresta, na trajetória do VANT.	148
8.2	Intervalo de confiança para a estimação da trajetória. . . . .	149
8.3	Transferência de estados por probabilidade condicional. . . . .	152
8.4	Diferença entre as medidas do GPS-RTK e INS. . . . .	154
8.5	Distribuições para amostragem no FP. . . . .	155
8.6	Aplicação do FP na estimação da posição do VANT. . . . .	156
8.7	Estimação da trajetória por FP. . . . .	157





## LISTA DE TABELAS

	<u>Pág.</u>
6.1	Parâmetros selecionados pelo MPCA para o primeiro experimento. . . . . 72
6.2	Incertezas em centímetros para cada coordenada - primeiro experimento. 72
6.3	Parâmetros selecionados pelo MPCA para o segundo experimento. . . . . 75
6.4	Incertezas em centímetros para cada coordenada - segundo experimento. 75
6.5	Incertezas em centímetros para cada coordenada - terceiro experimento. . 76
6.6	Erros médios de cada coordenada por experimento. . . . . 83
6.7	Incertezas em centímetros para cada coordenada, após remoção das partes ruidosas - quinto experimento. . . . . 90
6.8	Comparação entre EX4 e EX5. . . . . 90
6.9	Tamanho da janela móvel e diferenças para cada filtro. . . . . 95
6.10	Incertezas em centímetros para cada coordenada, após aplicação de filtros lineares - sexto experimento. . . . . 96
6.11	Comparação com EX4 e EX5. . . . . 97
6.12	Comparação com EX7 e EX5. . . . . 100
6.13	Comparação com EX8 e EX5. . . . . 101
7.1	Incertezas para navegação por visão computacional. . . . . 107
7.2	Incertezas para navegação por visão computacional, após filtragem. . . . 110
7.3	Incertezas para a fusão de dados. . . . . 116
7.4	Incertezas para a fusão de dados, após filtragem nos dados inerciais. . . . 119
7.5	Parâmetros selecionados pelo MPCA para o primeiro experimento. . . . . 122
7.6	Incertezas para a fusão de dados, com a implementação do fusor neural no computador <i>Zybo Z7</i> . . . . . 135
7.7	Incertezas para a fusão de dados, com a implementação do fusor neural no computador <i>Zybo Z7</i> e ativação CORDIC. . . . . 138
7.8	Incertezas para a fusão de dados, com a implementação do fusor neural no computador <i>Zybo Z7</i> e ativação LUT com interpolação. . . . . 142
8.1	Erros médios para a faixa de intervalo de confiança. . . . . 150
8.2	Incertezas para estimação da trajetória por FP. . . . . 157
9.1	Evolução dos resultados no Capítulo 6. . . . . 162
9.2	Evolução dos resultados no Capítulo 7. . . . . 164
9.3	Evolução dos resultados em <i>hardware</i> dedicado. . . . . 164
9.4	Resultados da estimação da posição por abordagem. . . . . 165





## LISTA DE ABREVIATURAS E SIGLAS

AEI & Anomalia Equatorial Ionosférica  
AMAS & Anomalia Magnética do Atlântico Sul  
ANAC & Agência Nacional de Aviação  
ARP & Aeronave Remotamente Pilotada  
AXI & Advanced Extensible Interface  
CI & Cintilação Ionosférica  
CLB & Configurable Logic Blocks  
CORDIC & Coordinate Rotation Digital Computer  
DECEA & Departamento de Controle do Espaço Aéreo  
DGPS & Differential Global Positioning Systems  
FAB & Força Aérea Brasileira  
FDP & Função Densidade de Propabilidade  
FK & Filtro de Kalman  
FP & Filtro de partículas  
FPGA & Field Programmable Gate Array  
GNSS & Global Navigation Satellite System  
GPS & Global Positioning System  
GPU & Graphics Processing Unit  
IEAv & Instituto de Estudos Avançados  
IEEE & Institute of Electrical and Electronic Engineers  
IMU & Inertial Measurement Unit  
INPE & Instituto Nacional de Pesquisas Espaciais  
INS & Inertial Navigation System  
LUT & Look up Table  
MLP & Field Multilayer Perceptron  
MLP & Multi Layer Perceptron  
MSMC & Método sequencial de Monte Carlo  
OACI & Organização de Aviação Civil Internacional  
OSHW & Open Source Hardware  
PITER & Processamento de Imagens em Tempo Real  
RAM & Random Access Memory  
RNA & Rede Neural Artificial  
RPAS & Remotely Piloted Aircraft System  
RTK & Real-time Kinematic  
SPI & Serial Peripheral Interface  
SoC & System on Chip  
UAS & Unmanned Aircraft Systems  
UAV & Unmanned Aerial Vehicle  
USB & Universal Serial Bus  
VANT & Veículo Aéreo não-tripulado  
VHDL & VHSIC Hardware Description Language  
VHSIC & Very High Speed Integrated Circuits  
WITAS & Wallenberg Laboratory for Information Technology and Autonomous Systems





## SUMÁRIO

	<u>Pág.</u>
<b>1 INTRODUÇÃO</b> . . . . .	<b>1</b>
1.1 Motivação . . . . .	3
1.2 Problema de pesquisa . . . . .	7
1.3 Objetivos . . . . .	10
1.4 Contribuições . . . . .	11
1.5 Estrutura da tese . . . . .	12
<b>2 REVISÃO BIBLIOGRÁFICA</b> . . . . .	<b>15</b>
2.1 Aplicação de VANT . . . . .	16
2.2 Estimação da posição por processamento de imagens . . . . .	18
2.3 Estimação da posição por fusão de dados . . . . .	20
2.4 Fusão de dados com redes neurais . . . . .	22
<b>3 FUNDAMENTAÇÃO TEÓRICA</b> . . . . .	<b>25</b>
3.1 Sensores GNSS e INS . . . . .	25
3.2 Navegação de VANT por imagens . . . . .	29
3.2.1 Estimação da posição do VANT por correlação de imagens . . . . .	31
3.2.2 Conversão de imagens para tons de cinza . . . . .	34
3.2.3 Filtragem em imagens . . . . .	36
3.2.4 Extração de bordas . . . . .	37
3.2.5 Extrator de bordas neural . . . . .	42
3.3 Fusão de dados . . . . .	44
3.3.1 Filtro de Kalman . . . . .	47
3.3.2 Redes neurais artificiais . . . . .	51
3.4 Desenvolvimento em FPGA . . . . .	52
3.4.1 Rede neural autoconfigurada . . . . .	55
<b>4 METODOLOGIA</b> . . . . .	<b>57</b>
4.1 Fases de experimentos . . . . .	61
<b>5 DADOS</b> . . . . .	<b>63</b>
5.1 Dados do WITAS . . . . .	63
5.2 Dados do IEAv . . . . .	65

<b>6</b>	<b>PRIMEIRA FASE DE EXPERIMENTOS - DADOS DO WITAS</b>	<b>69</b>
6.1	Treinamento e validação . . . . .	69
6.1.1	Treinamento com validação cruzada - <i>hold-out</i> . . . . .	71
6.1.2	Treinamento com validação cruzada - <i>k-fold</i> . . . . .	74
6.1.3	Considerações sobre a etapa de treinamento e validação . . . . .	82
6.2	Avaliação ao aplicar filtragem . . . . .	84
6.2.1	Remoção de ruído . . . . .	84
6.2.2	Aplicação de filtros lineares . . . . .	91
6.2.3	Considerações sobre a avaliação ao aplicar filtragem . . . . .	97
6.3	Combinação dos sensores de atitude . . . . .	98
6.3.1	Comparação entre acelerômetro e giroscópio . . . . .	98
6.3.2	Considerações sobre a combinação dos sensores de atitude . . . . .	101
<b>7</b>	<b>SEGUNDA FASE DE EXPERIMENTOS - DADOS DO IEAV</b>	<b>103</b>
7.1	Erro associado à navegação por visão computacional . . . . .	104
7.1.1	Estimação da trajetória por visão computacional . . . . .	106
7.1.2	Considerações sobre a estimação da trajetória por visão computacional	110
7.2	Inclusão dos dados inerciais . . . . .	114
7.2.1	Fusão de dados . . . . .	115
7.2.2	Considerações sobre a inclusão dos dados inerciais . . . . .	119
7.3	Desenvolvimento de um coprocessador em <i>hardware</i> dedicado . . . . .	122
7.3.1	Fusão de dados no computador <i>Raspberry pi</i> . . . . .	123
7.3.2	Fusão de dados no computador <i>Zybo Zynq 7000</i> . . . . .	131
7.3.3	Considerações sobre o desenvolvimento do coprocessador em <i>hardware</i> dedicado . . . . .	143
<b>8</b>	<b>QUANTIFICAÇÃO DE INCERTEZAS</b>	<b>147</b>
8.1	Estimação do desvio padrão . . . . .	147
8.2	Estimação da trajetória por filtro de partículas . . . . .	150
8.3	Considerações sobre a quantificação de incertezas . . . . .	158
<b>9</b>	<b>CONCLUSÃO</b>	<b>161</b>
9.1	Trabalhos futuros . . . . .	166
	<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>	<b>167</b>



# 1 INTRODUÇÃO

Ao longo da história da humanidade, as máquinas foram desenvolvidas para auxiliar o trabalho humano. Alguns escritores indicam a Revolução Industrial como marco inicial deste período, que dura até a atualidade. No decorrer do tempo, intensificou-se a utilização de máquinas de modo que fica difícil imaginar a vida humana sem a inclusão de dispositivos e tecnologias (STEARNS, 2020).

Com o aumento do número de máquinas, surge a demanda por aprimoramento e desenvolvimento das tecnologias utilizadas. Na atualidade, as máquinas chegam a substituir a mão de obra humana à medida que passam a ser usadas para além das tarefas para qual foram originalmente fabricadas (STEARNS, 2020). As máquinas que substituem a mão de obra humana são denominadas de robôs, que significa trabalhador, ou alguém que trabalha sem estafar (CORKE, 2011).

A ciência que estuda o desenvolvimento de robôs é denominada de robótica (NIKUS, 2013). Atualmente, os robôs são aplicados nas mais diversas áreas. Encontra-se na literatura, múltiplas denominações ou classificações para os tipos de robôs. Entretanto, é possível destacar duas classificações (CORKE, 2011; NIKUS, 2013):

- Dispositivo de Movimentação Manual: apresenta vários graus de liberdade, mas é controlado por um operador humano;
- Robô Inteligente: possui capacidade de compreender o ambiente no qual está inserido, detectar as possíveis mudanças e concluir com êxito a tarefa para qual foi programado/projetado.

Os dispositivos de movimentação manual são facilmente encontrados nas atividades humanas. Um exemplo são os automóveis, que não possuem autonomia para tomada de decisão, sendo completamente controlados por um operador humano. Por outro lado, os robôs inteligentes são máquinas projetadas para atuar na ausência ou com pouca interferência de um operador humano (CORKE, 2011). Atualmente, o estudo, o desenvolvimento e a aplicação de robôs inteligentes estão entre os principais temas investigados pela área da robótica (LEE et al., 2018; HUANG et al., 2019). Essas máquinas são denominadas de sistemas autônomos (ELECTRICAL; IEEE, 2020).

Pode-se definir um sistema autônomo como sistema capaz de desenvolver por si mesmo as leis, regras e/ou estratégias, que regulam, monitoram e modificam seu comportamento no ambiente onde eles estão inseridos (ELECTRICAL; IEEE, 2020).

Os sistemas autônomos executam suas tarefas sem interferência externa, adequando seu comportamento mesmo as situações imprevistas, o que é uma habilidade que é denominada de navegação autônoma (ELECTRICAL; IEEE, 2020).

A navegação autônoma pode ser aplicada à veículos aquáticos, terrestres e aéreos (DOMICIAN, 2018; SETIAWAN et al., 2019; GANDHI; SALVI, 2019). Em particular, destaca-se a navegação autônoma de veículos aéreos não tripulados, especialmente, em missões nas quais a presença de um piloto a bordo pode ser um fator limitante (HUTTUNEN, 2017; FUHRMANN; HOROWITZ, 2017; FIORI et al., 2017; ZHILENKOV; EPIFANTSEV, 2018). Comumente, a navegação autônoma de veículos aéreos é aplicada a aeronaves de pequeno porte não tripuladas (AL-KAFF et al., 2018).

Existe mais de uma denominação para aeronaves não tripuladas e de pequeno porte. Algumas denominações são: veículo aéreo não tripulado (VANT), aeronave remotamente pilotada (ARP), *unmanned aircraft systems* (UAS), *unmanned aerial vehicle* (UAV) ou ainda *Drone* - termo muito usado por órgãos de imprensa. O termo *Drone* significa zangão na língua inglesa, sendo uma autonomasia derivada devido ao barulho provocado por hélices durante o voo (LIMA, 2018). A Organização de Aviação Civil Internacional (OACI), que possui abrangência internacional, adota para esse tipo de aeronave não tripulada o termo *remotely piloted aircraft System* (RPAS) (LIMA, 2018). Doravante, neste trabalho, o termo VANT será empregado para identificar uma aeronave de pequeno porte não tripulada.

A Agência Nacional de Aviação Civil, define VANT em duas categorias. Elas são aeromodelos e ARP. Os aeromodelos são as aeronaves não tripuladas remotamente pilotadas usadas para recreação ou lazer e as ARP são as aeronaves não tripuladas utilizadas para outros fins como experimentais, comerciais ou institucionais (AGÊNCIA NACIONAL DE AVIAÇÃO CIVIL, 2020). Durante os últimos anos, pôde-se observar um contínuo aumento do emprego dos veículos aéreos, em particular, VANTs. As aplicações de VANTs acontecem em diversas áreas, que podem ser: atividades civis e militares, tais como operações de vigilância, monitoramento ambiental, aerelevantamento e transporte de carga. Particularmente, destaca-se a aplicação no monitoramento agrícola e agricultura de precisão (GEVAERT et al., 2014; DASH et al., 2017; DARYANAVARD; HARIFI, 2018; YANG et al., 2019; YAMAZAKI et al., 2019; RAO; RAO, 2019; GAO et al., 2020). Diante disso, entende-se que a aplicação de VANTs é algo consolidado em várias atividades.

## 1.1 Motivação

Devido o aumento do emprego de VANTs, cresceu também o interesse de empresas, organizações governamentais e da comunidade científica, no desenvolvimento de sistemas autônomos aéreos, principalmente de pequeno porte (YANG et al., 2019; GAO et al., 2020).

Em geral, a navegação de VANT é controlada por um piloto humano, que envia comandos por um sinal de rádio e pilota através de uma estação de solo (VALAVANIS; VACHTSEVANOS, 2014). A navegação automática é outra forma de navegação de VANT, que possibilita dispensar o controle do piloto humano por algum tempo à medida que este tipo de navegação é feita a partir de uma rota pré-estabelecida (VALAVANIS; VACHTSEVANOS, 2014). Já a navegação autônoma dispensa, completamente, o controle de um piloto humano.

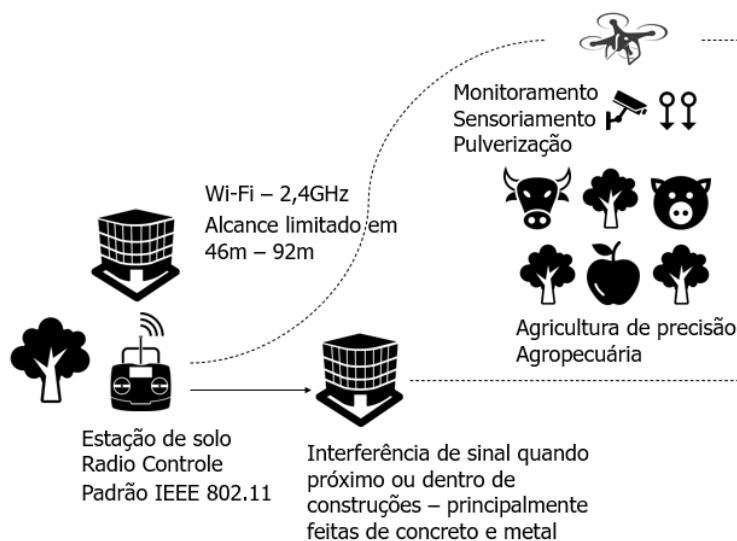
Tanto a navegação automática quanto a navegação autônoma necessitam de sensores, como um *Global Navigation Satellite System* (GNSS) e um *Inertial Navigation System* (INS) (VALAVANIS; VACHTSEVANOS, 2014). Tais sensores auxiliam no sistema de controle, na localização e posição do VANT durante a navegação (FRADEN, 2010; VALAVANIS; VACHTSEVANOS, 2014). Contudo, existem problemas associados aos sensores GNSS e INS. Os principais problemas são interferências, que podem ser provocadas intencionalmente ou por ação natural, o que pode inviabilizar a navegação autônoma. As principais formas de interferências intencionais são denominadas de *Jamming* e *Spoofing* (SILVA FILHO, 2016).

*Jamming* e *Spoofing* são ações intencionais tomadas para que um sinal seja introduzido em um canal de comunicação, na intenção de provocar erros ou como um meio de fornecer informações incorretas ao receptor do canal (GAO et al., 2020). No contexto da navegação autônoma de VANT, *Jamming* e *Spoofing* são ataques feitos sobre o receptor GNSS, embarcado no VANT, de modo a corromper o sinal GNSS ou inserir alguma informação incorreta (KIM et al., 2012; KIM et al., 2013; ELEZI et al., 2019; JIN et al., 2019; MEI et al., 2019; ARTHUR, 2019; GAO et al., 2020). Estes ataques afetam a estimação da localização e a posição do VANT, fornecida pelo receptor GNSS.

Além dos problemas com o sinal GNSS, deve-se considerar também as contingências com o INS, que é um sistema capaz de calcular a posição, velocidade e atitude do VANT (GREWAL et al., 2019). Este sistema é capaz de localizar o norte magnético pela rotação da Terra a partir de giroscópios e acelerômetros (GROVES, 2015). Com



Figura 1.2 - Alcance de tecnologias com o padrão IEEE 802.11 (Wi-Fi).



Fonte: Produção do autor.

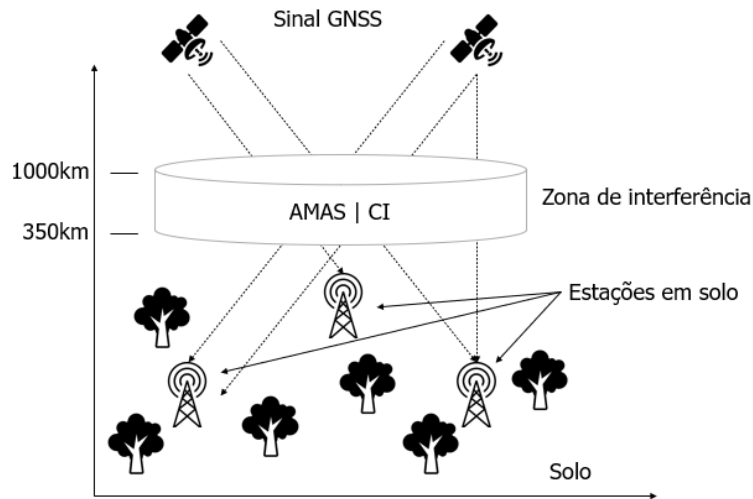
Além das interferências intencionais, também deve-se evidenciar as que são naturais. Como exemplo, destaca-se três fenômenos, que são a Anomalia Magnética do Atlântico Sul (AMAS), a Anomalia Equatorial Ionosférica (AEI) (BALAN et al., 2018) e a Cintilação Ionosférica (CI) (TAKAHASHI et al., 2001; CILLIERS et al., 2009; NASUDDIN et al., 2015). Efeitos da cintilação estão associados também a bolhas ionosféricas (TAKAHASHI et al., 2001).

A AMAS é a região de menor intensidade do campo magnético na superfície da Terra. Por isso, ela é uma região onde, potencialmente, a cintilação ionosférica pode ser mais intensa devido a precipitação de partículas de alta energia (CILLIERS et al., 2009). Este fenômeno incide sobre o território sul-americano, possibilitando que os efeitos mais evidentes sejam sentidos a partir de cem quilômetros de altitude. A AMAS afeta o funcionamento de satélites artificiais e pode afetar o sinal GNSS, provocando sua ausência (CILLIERS et al., 2009). A CI provoca uma variação rápida da amplitude e da fase dos sinais de ondas de rádio, o que ocorre quando estes sinais atravessam irregularidades ionosféricas ou bolhas de plasma. Estas irregularidades são provocadas por uma variação na densidade de elétrons na ionosfera, o que também afeta o sinal GNSS (CILLIERS et al., 2009; SUN et al., 2018; GULATI et al., 2019).

Os efeitos e as causas da CI e da AMAS são assuntos muito abordado no campo da geofísica espacial, visto que seu comportamento varia significamente, dependendo

da região terrestre observada (CILLIERS et al., 2009; NASUDDIN et al., 2015). Além disso, as técnicas empregadas para solucionar este problema ainda não conseguem a resolução completa. Elas não garantem o impedimento da interferência no sinal GNSS, em particular no sistema de posicionamento global norte americano (GPS) - este sistema é o mais utilizado no Brasil. A Figura 1.3, ilustra a zona de interferência, onde ocorre os fenômenos AMAS e CI. As interferências no sinal GNSS podem ocorrer a partir da altura de 350km (CILLIERS et al., 2009; SUN et al., 2018; GULATI et al., 2019).

Figura 1.3 - Cintilação Ionosférica e Anomalia Magnética do Atlântico Sul.



A interferência ocorre a uma altura entre 350 km e 1000 km

Fonte: Produção do autor.

Os problemas supracitados já foram discutidos e algumas soluções foram propostas para a navegação autônoma de VANTs, quando não há disponibilidade do sinal GNSS. Todas as soluções procuram garantir a qualidade da estimação da posição, quando ocorre ausência do sinal GNSS ou falha nas medidas do INS. Dessa forma, a solução indicada é a utilização da fusão de dados e/ou navegação por processamento de imagens (STAMATESCU et al., 2015; CORRADO; PANETTA, 2017; BRAGA, 2018; DOMICIAN, 2018; PAULINO, 2019; REBOUÇAS, 2019; KHALDI et al., 2019; ZHANG; XU, 2019).

Os estudos sobre navegação autônoma de VANTs por processamento de imagens, mostram resultados promissores. Principalmente, quando não existe uma dependência do sinal GNSS para estimar a posição e localização do VANT, durante a

navegação. Contudo, ainda existem problemas associados a navegação autônoma por processamento de imagens. Tais problemas advêm da sua natureza técnica, que é sensível à interferências de fatores ambientais, como nuvens, chuva, fumaça, luminosidade, sensores imageadores de baixa qualidade ou qualquer outro fator que atrapalhe a coleta de imagens durante a navegação (SILVA, 2016; SILVA FILHO, 2016; BRAGA, 2018). Portanto, as abordagens que utilizam imagens, para navegação autônoma de VANTs, ainda não são totalmente eficazes.

Essencialmente é importante garantir confiabilidade, precisão e robustez, para a navegação autônoma de VANTs. Por isso, para mitigar possíveis falhas das alternativas, que utilizam apenas imagens, a solução sugerida na literatura é aplicar técnicas de fusão de dados. Com a fusão de dados, é possível obter maior confiabilidade à navegação autônoma de VANTs e, desta forma, garantir uma melhor estimativa da posição do VANT durante o voo (KHALEGHI et al., 2013; AL-KAFF et al., 2018; KHALDI et al., 2019; ZHANG; XU, 2019).

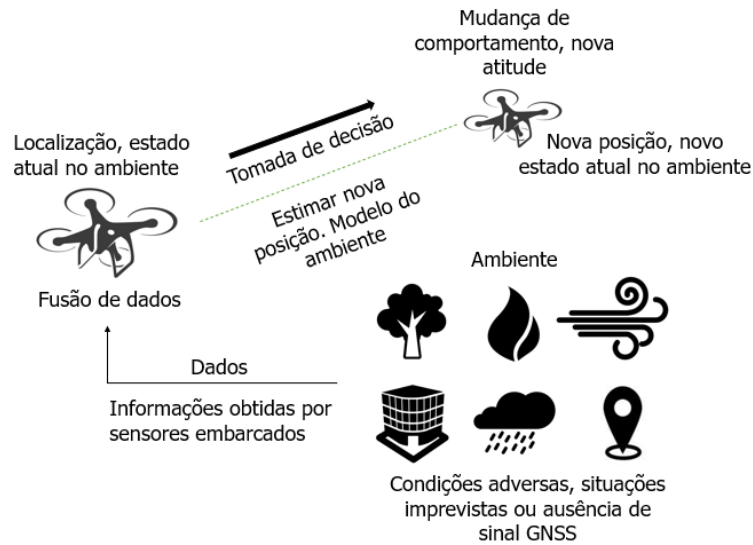
## 1.2 Problema de pesquisa

A fusão de dados opera com detecção, associação, correlação, estimação e combinação de diferentes tipos de dados em sensores diversos (KHALEGHI et al., 2013). Neste sentido, a fusão de dados torna-se uma alternativa interessante para aplicação na navegação autônoma de VANTs. Quanto mais sensores e tipos de dados combinados, aumenta-se a quantidade de informação usada nas estimações. A partir disso, pode-se melhorar a estimação feita pelo sistema fusor, durante a ausência do sinal GNSS (SILVA FILHO, 2016; PAULINO, 2019). A Figura 1.4, ilustra o processo de navegação autônoma de um VANT, por fusão de dados.

Contudo, existem problemas associados a fusão de dados. Destaca-se, por exemplo, que utilizar diversos sensores pode implicar em ter que processar grandes volumes de dados, o que exige maior custo computacional (PAULINO, 2019).

Isso pode inviabilizar o embarque do fusor em VANTs de pequeno porte, que não possuem um computador de bordo robusto suficiente para processar a fusão de dados em tempo hábil. Além disso, quanto mais sensores são usados, aumenta-se a possibilidade de fontes de dados ruidosas, com características particulares, taxas de amostragem descasadas ou ainda de natureza e princípios físicos diferentes (OPANASIUK, 2018; ELECTRICAL; IEEE, 2020). Diante disso, é necessário que a técnica usada como fusor de dados, seja capaz de correlacionar dados oriundos de diferentes fontes, com características particulares.

Figura 1.4 - Processo de navegação autônoma, por fusão de dados, para um VANT.



Fonte: Produção do autor.

Entre as técnicas que se destacam, como fusor de dados, é o Filtro de Kalman (FK) (CONTE; DOHERTY, 2009; KONG et al., 2019; YOU et al., 2020). O FK é considerado uma técnica clássica, aplicada a fusão de dados e navegação autônoma de VANTs (PAULINO, 2019).

Contudo, o FK possui desvantagens. Entre elas, as que mais se sobressaem são:

- Em grande parte dos trabalhos que utilizam o FK, o ruído no sistema de navegação do VANT é assumido como gaussiano, de média zero. Em contrapartida, não é sempre que o tipo de ruído obtido pelo sistema de navegação do VANT é gaussiano, o que pode provocar imprecisões nas estimações do FK. Além disso, é necessário informar a priori as matrizes de covariância dos ruídos (WANG; WANG, 2019);
- A complexidade exigida na construção/implementação do FK é alta. Além disso, o custo computacional de executar o FK em VANTs, de pequeno porte, pode virar um problema se não houver um computador de bordo capaz de suportar a execução do FK, em tempo hábil para fusão de dados (KONG et al., 2019);
- A linearização do sistema de navegação do VANT pode causar divergências nas estimações do FK (ZHANG; XU, 2019).



Além do FK, existem outras técnicas que podem ser usadas na fusão de dados, para navegação autônoma de VANTs (ADUSUMILLI et al., 2015; CORRADO; PANETTA, 2017; PAULINO, 2019; ZHANG; XU, 2019).

Geralmente, as técnicas para substituir o FK, são desenvolvidas com base em algum algoritmo de aprendizagem de máquina, com o objetivo de reduzir as desvantagens do FK. Contudo, os estudos que demonstram essas abordagens, não as executam em ambientes de processamento de alto desempenho, como *Field Programmable Gate Array* (FPGA) (PEDRONI, 2008; PIMENTA, 2017). Isto implica na necessidade de usar uma ou mais CPUs para executar as técnicas sugeridas, o que pode não ser viável em VANTs de pequeno porte.

Outra questão é que, apesar de algumas técnicas possuírem complexidade de implementação menor que o FK, não é possível afirmar que elas possuem um desempenho igual ou superior ao FK com relação a velocidade de execução. Essa impossibilidade ocorre devido a variação de acordo com a velocidade do *hardware* embarcado no VANT.

Diante desses problemas, esta Tese busca aplicar uma alternativa de menor complexidade computacional e de implementação, quando comparada com o FK (NOWOSAD, 2001).

Além disso, esta alternativa deve permitir sua execução em ambientes de alto desempenho computacional, como FPGA. Assim, é possível garantir a redução do custo computacional ao mesmo tempo que se tem uma velocidade de processamento igual ou superior as técnicas executadas em CPU. A alternativa investigada neste trabalho é a Rede Neural Perceptron Multicamada, do inglês *Multi Layer Perceptron* (HAYKIN, 2009). Contudo, este trabalho utilizou uma MLP autoconfigurada (ANOCHI et al., 2015).

O algoritmo utilizado para se obter a MLP autoconfigurada é o MPCA, que é uma metaheurística (ANOCHI et al., 2015). Existe um ponto que precisa ser considerado sobre a aplicação do MPCA, que é a forma de seleção dos dados usados para treinamento da rede neural. Ressalta-se que o MPCA não foi desenvolvido, especificamente, para o problema da navegação autônoma de VANT. Logo, torna-se necessário um estudo de como selecionar os dados de treinamento da MLP autoconfigurada. Nesta Tese, utilizou-se técnicas de validação cruzada para aprimorar a seleção dos dados de treinamento, o que permite melhorar os resultados da fusão de dados com a rede neural autoconfigurada. Além das técnicas de validação cruzada, também

buscou-se avaliar a combinação dos dados dos sensores de altitude, acelerômetros e giroscópios (WEI et al., 2018). Os sensores indicados apresentam informações complementares sobre o posicionamento da aeronave no espaço. Portanto, caso seja necessário, é possível desprezar uma destas informações de acordo com a necessidade de apontar qual é a mais relevante para o conjunto de dados de treinamento.

Outro problema evidenciado na literatura, é a presença de ruído tanto nos dados inerciais como nas imagens coletadas durante o voo (CONTE; DOHERTY, 2009; SILVA FILHO, 2016; WANG; WANG, 2019; YOU et al., 2020). Neste trabalho, são aplicadas técnicas de filtragem para avaliar o impacto provocado pela remoção de ruído.

Considera-se que a estimação da incerteza da fusão de dados, durante a navegação autônoma do VANT, também é um problema pouco abordado na literatura. Assim como as técnicas baseadas em processamento de imagem, a fusão de dados também pode apresentar falhas, ou seja, pode gerar estimações da posição do VANT distantes do que seria ideal para navegação autônoma. Nesse contexto, torna-se necessário entregar uma maneira de informar o computador de bordo, responsável pela navegação autônoma, sobre a eficiência da estimação da posição. Isto possibilita tomadas de decisão por parte do computador de bordo que garantem mais segurança, como um pouso rápido para não permitir o distanciamento do VANT de sua rota. Neste trabalho, é apresentada uma proposta para estimar a incerteza da fusão de dados, durante o voo. A proposta também usa uma MLP autoconfigurada.

Por fim, a falha conjunta do sinal GNSS e do equipamento que transporta o sensor imageador é uma situação crítica que pode ocorrer durante a navegação autônoma de VANT. Nesta situação, o computador de bordo deixa de receber duas informações importantes para navegação, o que torna necessário aplicar uma alternativa que permita a navegação, pelo menos até o retorno do sinal GNSS. Nesta Tese, a alternativa investigada foi estimar a posição do VANT por um Filtro de Partículas (SCHOTT et al., 2017; PETUKHOV et al., 2019; GUANG; JI, 2020).

### 1.3 Objetivos

O presente trabalho tem como principal objetivo investigar, analisar e qualificar a estimação da posição de um VANT. Isso será desenvolvido a partir da fusão dos dados, dos sensores embarcados: sensor inercial e câmeras. No processo, não é utilizado sinal de posicionamento por satélites. Técnica de processamento de imagem é usada para estimar o posicionamento do VANT, onde uma MLP autoconfigurada é utilizada para extração de bordas. A fusão de dados para posicionamento do VANT

é calculada por uma segunda rede MLP autoconfigurada, como alternativa ao filtro de Kalman.

Além de fornecer uma alternativa ao FK, o objetivo também é a construção de um dispositivo de processamento em alto desempenho, um *hardware* dedicado. Dessa forma, será possível implementar a técnica investigada num dispositivo pequeno e possível de embarcar em VANT de pequeno porte. Outro objetivo que pode ser assinalado é analisar o impacto sobre as estimações a partir da fusão de dados feita em *hardware* dedicado.

Pode-se destacar outros objetivos secundários como:

- Aumentar a segurança de voo, possibilitando a implantação de sistemas de posicionamento redundantes para a navegação autônoma;
- Desenvolver uma abordagem baseada em inteligência artificial - precisamente com MLP autoconfigurada e com arquitetura otimizada;
- Analisar a fusão de dados sobre dados de voos reais e não sobre dados simulados.

#### 1.4 Contribuições

Neste trabalho, é aplicado a junção de duas técnicas de estimação da posição de VANTs para navegação autônoma. A primeira é a estimação da posição por processamento de imagens. A segunda é a estimação da posição por fusão de dados, com uma MLP autoconfigurada.

O trabalho foi dividido em fases e cada fase em etapas. Para cada etapa buscou-se responder um ou mais questionamentos. Dessa forma, as contribuições geradas são:

- Investigação de uma técnica de fusão de dados, baseada em uma MLP autoconfigurada, para substituir a técnica clássica, baseada FK;
- Treinamento e avaliação de MLP autoconfigurada, utilizando duas técnicas clássicas de validação cruzada;
- Desenvolvimento de uma nova abordagem de seleção de dados e treinamento, com validação cruzada, para uma MLP autoconfigurada.

- Análise do impacto que a forma de seleção dos dados de treinamento provoca nas estimações feitas pela MLP autoconfigurada;
- Aplicação e análise de resultados de mais de uma técnica de filtragem, tanto em dados inerciais como em imagens;
- Avaliação de quais sensores de atitude, quando combinados com os demais sensores, produzem o melhor resultado para estimação da posição por fusão de dados com a MLP autoconfigurada;
- Desenvolvimento de um coprocessador em *hardware* dedicado, com FPGA. Implementação em dois tipos de dispositivos, para embarque em VANTs de pequeno porte;
- Análise do impacto, nas estimações da posição do VANT, quando a fusão de dados é feita com FPGA;
- Investigação de três abordagens para construção da função de ativação, da MLP autoconfigurada, em FPGA e análise dos resultados para cada abordagem;
- Quantificação de incertezas, em tempo real, durante a navegação autônoma com uma MLP autoconfigurada;
- Aplicação de um Filtro de Partículas para estimação da posição de um VANT, a partir do desvio padrão obtido pela diferença entre medidas do INS e de um equipamento GPS de alta precisão.

## 1.5 Estrutura da tese

A texto desta Tese, está dividido da seguinte forma:

- Capítulo 2, apresenta a revisão bibliográfica que sustenta o desenvolvimento desta Tese. Além da revisão bibliográfica, é abordado um breve histórico sobre a aplicação de VANT;
- Capítulo 3, descreve o fundamentação teórica que foi base para o desenvolvimento do trabalho, bem como os principais métodos utilizados;
- Capítulo 4, descreve a metodologia proposta para navegação autônoma de VANT por fusão de dados com MLP autoconfigurada. Neste capítulo, é explicado como cada parte da metodologia deve ser integrada para construir o fusor neural;

- Capítulo 5, apresenta os dados utilizados nos experimentos e descreve como estes dados foram obtidos;
- Capítulo 6, apresenta a primeira fase de experimentos. Neste capítulo, é investigado se a MLP autoconfigurada consegue emular o FK, como fusor de dados. Os estudos abordados versam sobre como selecionar os dados de treinamento, validação e generalização para aprimorar o treinamento da MLP, além de discutir o impacto de ruído nos dados;
- Capítulo 7, apresenta a segunda fase de experimentos. Neste capítulo, são mostrados os experimentos no Brasil. Também é feita uma análise sobre os resultados das etapas de estimação da posição do VANT. É neste capítulo que o *hardware* dedicado é explicado e testado;
- Capítulo 8, apresenta uma abordagem para estimação de incertezas durante o voo autônomo. Também mostra a aplicação do Filtro de Partículas para estimação da posição do VANT;
- Capítulo 9, apresenta as conclusões para cada etapa de experimentos e as considerações finais sobre os resultados obtidos nesta Tese.



## 2 REVISÃO BIBLIOGRÁFICA

A fabricação de VANTs não é recente, apesar do uso deste tipo de aeronave ter se popularizado nos últimos anos. Como muitas tecnologias usadas atualmente, seu surgimento aconteceu devido a demandas militares. O primeiro desenvolvimento de VANT começou logo após o primeiro voo tripulado em 1916, quando a empresa Sperry Gyroscope desenvolveu um sistema de controle automático para o hidroavião Curtiss (LAYMAN, 1996).

A primeira aeronave considerada remotamente pilotada foi a F-51 (KEANE; CARR, 2013; BLOM, 2010). Ela foi projetada e testada pela marinha norte americana em 1924. A construção da F-51 foi possível devido aos avanços da aviação comercial, durante o período entre guerras, de 1918 a 1939. Neste período, ocorreu o aprimoramento de diversas tecnologias, dentre elas, o sistema de rádio controle (KEANE; CARR, 2013).

No passado, uma das primeiras aplicações de VANTs foi em projetos de bombas aéreas (LAYMAN, 1996). Além do transporte de bombas, outra aplicação de VANTs foi em missões de espionagem e inteligência (KEANE; CARR, 2013). Durante a Guerra do Vietnã, entre os anos 1955 e 1975, foi utilizado um VANT denominado *Lightning Bug*. Este VANT era equipado com uma câmera de alta qualidade, para a época, utilizada para capturar imagens das tropas vietnamitas (LAYMAN, 1996).

Na década de 1990, foi projetado o VANT AAI RQ-2 Pioneer, que foi considerado um marco no desenvolvimento de VANTs (BLOM, 2010). Este VANT foi desenvolvido a partir de uma cooperação entre Israel e EUA e foi utilizado em mais de 300 missões, entre os anos de 1990 e 1991, na guerra do Golfo (BLOM, 2010). Mesmo com o fim da guerra do Golfo, o AAI RQ-2 Pioneer continuou sendo utilizado em outros conflitos como na Guerra da Bósnia, Guerra da Somália e, mais recentemente, na Guerra do Afeganistão, entre Outubro de 2001 e Dezembro de 2014 (KEANE; CARR, 2013).

No final da década de 90, a despeito das aplicações militares, começou-se a utilizar VANTs em aplicações civis. Isto foi possível devido ao barateamento nos custos de produção de aeronaves de pequeno porte e, também, uma miniaturização de componentes eletrônicos. Isso permitiu embarcar tecnologias para ampliar a utilização dessas aeronaves, dentre estes componentes estão os sensores de geolocalização e sensores imageadores (BLOM, 2010; KEANE; CARR, 2013).

## 2.1 Aplicação de VANT

Atualmente, torna-se indiscutível a tendência mundial no emprego de VANTs. Há diversas aplicações debatidas na literatura. O principal exemplo encontrado são as aplicações em que a presença humana deve ser evitada ou não pode ser utilizada (RAO; RAO, 2019).

Algumas aplicações são pensadas para problemas em que a mão de obra, exclusivamente humana, gastaria muito tempo para concluir uma tarefa. Por exemplo, no trabalho de CHIROLI et al. (2017), é apresentado um estudo com o objetivo de avaliar a viabilidade do uso de VANT, no auxílio ao programa de controle da dengue executado no Brasil, na cidade de Maringá - PR. Outro exemplo, está no trabalho de SANDINO et al. (2018), onde é apresentado um método que integra VANTs, sensores de imagem hiperespectral e algoritmos de processamento de dados, usando aprendizado de máquina, para o mapeamento aéreo de florestas afetadas por patógenos.

Dentre as aplicações de VANTs, destaca-se as que se envolvem no agronegócio e na conservação ambiental. No trabalho de GEVAERT et al. (2014), é apresentado um estudo que usa imagens de VANT com o objetivo de monitorar o crescimento da cultura em um campo de batata. No trabalho de (DASH et al., 2017), é apresentado que VANTs podem fornecer novas ferramentas para melhorar o monitoramento da saúde florestal, fornecendo dados com resoluções temporais e espaciais muito altas.

Ainda, com relação ao agronegócio e conservação ambiental, no trabalho de BAENA et al. (2017), é apresentado um estudo com aplicação de um VANT em projetos de conservação de plantas que vão desde a vegetação hiperárida do Peru até as florestas secas do Caribe e, finalmente, a floresta úmida da África do Sul e da Amazônia brasileira. No trabalho de RAO e RAO (2019), é apresentado um projeto, em que um VANT é usado para pulverização agrícola, que pode ser feita de forma semi-autônoma. O objetivo do projeto é evitar o contato direto dos agricultores com pesticidas para evitar problemas de saúde.

YINGKUN (2018) apresenta um algoritmo para o planejamento da trajetória de voo de um VANT agrícola. Ele estuda como gerar o planejamento de voo de modo a evitar obstáculos. No trabalho de MAES e STEPPE (2018), é revisado o progresso do sensoriamento remoto com VANTs no: estresse hídrico, detecção de ervas daninhas e patógenos, avaliação do estado nutricional e vigor de crescimento e na previsão de produção agrícola.



A aplicação de VANTs estende-se para além do agronegócio e conservação ambiental, os estudos iniciam que VANTs também podem ser aplicados a estudos na biologia, como, por exemplo, no trabalho de [FIORI et al. \(2017\)](#), que aborda a aplicação de VANTs como uma ferramenta de pesquisa para estudos de mamíferos marinhos.

Outra aplicação importante para VANTs acontece em missões de resgate. No trabalho de [YAMAZAKI et al. \(2019\)](#), é apresentado um sistema de detecção de vítimas baseado em reconhecimento de voz, usando um VANT. O objetivo do trabalho é atender situações de busca, localização e resgate, quando ocorrer um desastre de grande escala, como um grande terremoto ou situações que existam vítimas soterradas.

Além das aplicações em resgate, há trabalhos que demonstram o uso de VANTs para monitoramento. No trabalho de [DARYANAVARD e HARIFI \(2018\)](#), é apresentado um sistema para detecção e reconhecimento de rostos, utilizando VANTs de pequeno porte. O objetivo do trabalho é identificar pessoas perdidas ou pessoas que estão foragidas. Além do monitoramento de pessoas, os VANTs também podem ser aplicados ao monitoramento de atividades. Isso é mostrado, por exemplo, no trabalho de [XIANG et al. \(2017\)](#), que descreve uma metodologia de monitoramento usando um sistema de monitoramento de sensoriamento remoto baseado em uma plataforma de VANT. Esta plataforma foi desenvolvida para o monitoramento da descargas térmicas em usina nucleares.

Ainda com relação ao monitoramento, os VANTs são aplicados para monitoramento de linhas e transmissão de energia. No trabalho de [MARTINS et al. \(2020\)](#), os autores analisam a aplicação de VANT em inspeções regulares e ocasionais de linhas de transmissão de energia. O objetivo é tornar esta atividade autônoma, mais rápida, segura e menos onerosa que os modos tradicionais. Como resultado, identificou-se melhorias em segurança, processo computacional e eficiência energética com baixo custo.

As aplicações de VANTs em situações militares continuam existindo. Por exemplo, no trabalho de [GAO et al. \(2020\)](#), estuda-se a interferência eletromagnética sofrida por VANTs, em um campo de batalha. O trabalho propõe um método de reconhecimento da situação de segurança de interferência eletromagnética, para que o VANT não saia do controle ou seja abatido.

Os trabalhos supracitados denotam, de maneira clara, porque há grande interesse na aplicação de VANTs. As possibilidades são muitas, principalmente quando se pensa

na navegação autônoma. Contudo, a navegação autônoma ainda é um desafio e, por isso, precisa de atenção e mais pesquisas. Assim, ela pode se tornar uma solução aplicada em várias atividades em que a presença humana não pode ser usada ou não é adequada (AL-KAFF *et al.*, 2018).

## 2.2 Estimação da posição por processamento de imagens

Devido à crescente demanda pela aplicação de VANTs, aumentou-se também os estudos que buscam soluções para a navegação autônoma. Dentre eles, destacam-se os que propõe a estimação e/ou correção da posição do VANT, por processamento de imagens (AL-KAFF *et al.*, 2018).

De maneira simples, grande parte dos métodos usados na estimação da posição de VANT, por processamento de imagens, utilizam o conceito de registro de imagens. O registro de imagens é o processo de sobreposição de duas imagens de uma mesma cena, capturadas em instantes de tempo diferentes e/ou sob um diferente ponto de vista, utilizando diferentes sensores (BRAGA, 2018).

O uso de processamento de imagens, como uma alternativo ao GNSS, possui algumas vantagens. As vantagens que mais se destacam são: esta abordagem não depende de um sinal externo, ou seja, sua aplicação acontece em tempo real e dentro de um computador embarcado no VANT. Outra vantagem é que os algoritmos de visão computacional usam principalmente sensores imageadores passivos, embarcados, que são, geralmente, mais baratos do que a manutenção de um GNSS (SILVA FILHO, 2016).

Os trabalhos, que investigam o uso de imagens para navegação autônoma de VANTs, sugerem metodologias que não dependam do GNSS e também mitigam o erro de deriva, que pode acontecer nas medidas do INS. Dentre estes trabalhos, pode-se comentar sobre o trabalho de SILVA FILHO (2016), que desenvolveu um método para navegação de VANTs através da identificação de marcos em imagens aéreas. Esta identificação baseou-se em técnicas de reconhecimento de objetos por pontos característicos.

Existe mais de uma forma, ou método, para aplicar o processamento de imagens na estimação da posição de VANTs. Por exemplo, no trabalho de URZUA *et al.* (2017), é proposto um sistema de localização e mapeamento simultâneo monocular. O sistema proposto é baseado em análises de observabilidade e foi desenvolvido para implementação em *hardware* de baixo custo.

No trabalho de [VIEIRA \(2019\)](#), é apresentada uma metodologia para o planejamento automático de rotas em VANTs, que utilizam um sistema de navegação por imagens baseado no reconhecimento de marcos.

No trabalho de [LIMA \(2018\)](#), é avaliado a acurácia do cálculo do posicionamento e a orientação de ARP. Para isto, são utilizados os conceitos de fotogrametria, a partir de imagens geradas por câmeras embarcadas, visando à navegação aérea autônoma.

No trabalho de [BRAGA et al. \(2019\)](#), são usadas duas técnicas de processamento de imagens para estimação da posição de um VANT, que são a odometria visual e a visão computacional. O posicionamento por odometria visual e visão computacional são combinados por meio de uma nova formulação de filtro de partículas, denominado filtro de partículas não extensivo.

O processamento de imagens feito pela abordagem de registro de imagens, não é trivial. Isto porque quando se combina as características do sensor imageador com as condições do ambiente, durante a aquisição das imagens, podem ocorrer falhas. Estas falhas são denotadas pela presença de ruídos nas imagens, como imagens borradas. Desta forma, torna-se necessário a aplicação de técnicas para tratamento das imagens obtidas durante o voo. Por exemplo, no trabalho de [WANG e HSIEH \(2018\)](#), é avaliado um método de aprimoramento de imagem não linear, para uma imagem borrada obtida por VANT.

Outro exemplo, que investiga a complexidade do processamento de imagens, está no trabalho de [LEE et al. \(2019\)](#). Os autores investigam o desempenho de algoritmos de odometria visual inercial e usam um software de simulação fotorrealista para obter a dinâmica da aeronave e, com isso, avaliar o desempenho das técnicas de odometria.

Além dos problemas com relação a complexidade e aquisição, também há problemas relacionados ao ambiente. Por exemplo, no trabalho de [SILVA \(2016\)](#), é apresentado um método capaz de estimar a posição geográfica de um VANT durante um voo noturno. Ela usa imagens aéreas no infravermelho que são capturadas pela aeronave, buscando a região sobrevoada em uma imagem de satélite georreferenciada na banda do visível.

No trabalho de [BRAGA \(2018\)](#), é apresentada uma metodologia que estima a posição geográfica de um VANT, quando este sobrevoa regiões cobertas por água e regiões com baixa ou sem luminosidade, utilizando imagens provenientes de um sensor ativo denominado *Light Detection And Ranging* (LiDAR).

Outro trabalho importante é o de [VITOR TORRES et al. \(2020\)](#). Neste trabalho os autores demonstram o uso de uma rede neural sem pesos, aplicada a vigilância de desmatamento da floresta amazônica. Um dos objetivos do trabalho é a navegação visual de Veículos Aéreos Não Tripulados. Neste trabalho, a extração de bordas nas imagens é realizada pela rede neural sem pesos de conexão, onde um conjunto de discriminadores são empregados na rede para segmentação das imagens.

Os trabalhos supracitados evidenciam que há vários pontos a se considerar sobre a aplicação de imagens para estimação da posição de VANTs. Qualquer solução proposta para a navegação autônoma por imagens deve levar em consideração aspectos relacionados com os processos de aquisição da imagem, como, por exemplo, a velocidade do obturador da câmera, a luminosidade da cena, o ambiente da cena, a escala, rotação e fatores projetivos, o sensor da câmera, etc. Além disso, aspectos que podem trazer restrições a qual método será mais eficaz estão relacionados com: o próprio algoritmo, a sua complexidade, o computador de bordo que irá processá-lo e a carga útil da aeronave. Desta forma, entende-se que não há uma solução única e que devido às várias situações que podem ocorrer na aquisição e processamento das imagens, torna-se interessante a junção e/ou seleção de mais de uma técnica ([SILVA FILHO, 2016](#); [FORNARI et al., 2018](#)).

### **2.3 Estimação da posição por fusão de dados**

Dado os problemas associados a estimação da posição de VANT por processamento de imagens, as pesquisas voltaram-se para investigação de alternativas que minimizam estes problemas. Então, uma abordagem que se destaca é a fusão de dados multisensor ([CONTE; DOHERTY, 2009](#); [KHALEGHI et al., 2013](#)).

Na fusão de dados, considera-se a informação oriunda de várias fontes, vários sensores embarcados. Estas informações são reunidas e fusionadas, ou seja, processadas por algum algoritmo, que é responsável por disponibilizar uma estimação da posição do VANT, considerando as várias informações ([GROVES, 2015](#)). Por exemplo, no trabalho de [CORRADO e PANETTA \(2017\)](#), foi demonstrado um exemplo de alteração de um VANT comercial já existente para avaliar desastres envolvendo radiação, combinando detecção de radiação, visão computacional e um GNSS.

No trabalho de [P. e KHANDEKAR \(2019\)](#), foi demonstrado a aplicação de um sistema que processa as informações da *Inertial Measurement Unit* (IMU). Neste sistema, são fusionadas as informações de seis eixos, em que três são para aceleração na direção X, Y, Z e eixo do giroscópio em X, Y, Z. Eles fornecem a taxa de guinada,

taxa de inclinação e taxa de rotação.

No trabalho de [BAO et al. \(2017\)](#), é apresentado um modelo aerodinâmico, para o INS. Este modelo consiste de um método de navegação tolerante a falhas de GPS para VANTs multirotores, baseado em *Federated Kalman Filter*.

Existem várias técnicas de fusão de dados, mas, muitas são baseadas em FK. Por exemplo, no trabalho de [TAN et al. \(2018\)](#), é proposto um algoritmo de fusão de dados. Ele é baseado em filtro complementar explícito e FK, usando controlador de piloto automático de código aberto para realizar o voo de um VANT em um ambiente fechado.

Contudo, demonstra-se na literatura que o FK tem desvantagens. Entre elas, está o seu custo computacional e também a complexidade para implementação do FK. No trabalho de [ZHANG e XU \(2019\)](#), é investigado o custo computacional do FK e uma nova técnica para fusão de dados é sugerida. Esta técnica é baseada no algoritmo *quaternion gradient descent* (QGD).

Com o objetivo de evitar a complexidade e custo computacional do FK, as pesquisas sobre fusão de dados sugerem outras abordagens. Geralmente, estas abordagens são baseadas em aprendizagem de máquina. Por exemplo, no trabalho de [KHALDI et al. \(2019\)](#), é apresentado uma forma de fusão de dados, visuais e inerciais, adquiridos usando uma técnica de fluxo óptico, em que o método de Lucas e Kanade é usado para estimar o deslocamento de pontos (*pixels*), representando linhas selecionadas em uma cena. Já no trabalho de [ADUSUMILLI et al. \(2015\)](#), é proposto um método híbrido de regressão de componentes principais e *Random Forest Regression* (RFR) para fusão de dados INS e GPS.

No trabalho de [KONG et al. \(2019\)](#), é proposta uma abordagem com custo computacional baixo, para navegação autônoma de VANTs, em ambientes com GPS negado. Os autores sugerem uma arquitetura de fusão de dados, em que os dados de um sensor primário e de um sensor secundário são fundidos. Quando o sensor secundário está indisponível na maior parte do tempo, a navegação do VANT usa a saída do sensor primário. Quando o sinal do sensor secundário está disponível, o sensor primário é realinhado, usando o sinal do sensor secundário para ligar os erros. Na abordagem dos autores, o sensor primário é a IMU e as entradas do sensor secundário são de um sistema de banda ultralarga.

No trabalho de [WANG e WANG \(2019\)](#), é apresentado o método *nonlinear signal-*

*correction observer*. Este método visa rejeitar erros grandes de medidas de posicionamento e estimar velocidades desconhecidas, utilizando integração de sinais e um controlador dinâmico para a posição e outro para a atitude. A técnica é capaz de minimizar erros, apesar da existência de ruídos não-gaussianos, que são filtrados por técnicas que envolvem a análise de robustez no domínio da frequência a fim de eliminar ruídos de alta frequência.

No trabalho de [PAULINO \(2019\)](#), é proposta uma aplicação de fusão de dados multisensorial de baixo custo, que é baseada em inteligência computacional híbrida adaptativa. O autor mostra uma melhora significativa na estimação da posição de um VANT, quando comparado ao uso de GPS/INS.

Todos os trabalhos supramencionados buscam uma alternativa para melhorar a precisão das estimações da posição de um VANT. Em particular, alguns trabalhos buscam evitar a complexidade do FK. Contudo, poucos trabalhos investigam o desempenho computacional de suas técnicas, ou seja, a velocidade de processamento da fusão de dados. Além disso, eles não sugerem uma alternativa para realizar a fusão de dados em um ambiente de alto desempenho, embarcado no VANT. Alguns trabalhos, inclusive, mostram em seus resultados que utilizaram CPUs para realizar a fusão de dados ([PAULINO, 2019](#)). CPUs que não foram embarcadas em VANTs.

Dado que a velocidade de processamento da fusão de dados é um fator crítico, torna-se necessário buscar abordagens que possam ser implementadas em dispositivos de processamento de alto desempenho, como FPGA ([PEDRONI, 2008](#); [PIMENTA, 2017](#)). A abordagem, que é considerada nesta Tese, parte da utilização de redes neurais artificiais ([SAMBATTI et al., 2016](#)).

## **2.4 Fusão de dados com redes neurais**

O deslocamento de um VANT durante o voo é algo que acontece em frações de segundo ([YINGKUN, 2018](#)). Portanto, as técnicas de fusão de dados devem processar informações com máxima eficiência, ou seja, na maior velocidade possível. Para solucionar esta questão e também de reduzir a complexidade computacional das técnicas baseadas em FK, nesta Tese, utilizou-se uma rede neural artificial do tipo MLP. Ela foi autoconfigurada para realizar a fusão de dados, com o objetivo de reduzir a complexidade computacional e construir um *hardware* dedicado a partir de uma MLP de topologia minimizada.

O uso de redes neurais costuma ser aplicado em diversos problemas ([HAYKIN, 2009](#)).

Por isso, pode-se considerar redes neurais como um conjunto de técnicas já consolidadas, mas que estão em constante evolução/adaptação (NIAN; JUNG, 2019). As redes neurais são modelos matemáticos, que foram projetados para delinear como o cérebro humano realiza uma tarefa de interesse (HAYKIN, 2009). A principal propriedade de uma rede neural é conseguir aprender a partir dos dados de entrada e, assim, melhorar seu desempenho.

A MLP é um modelo de rede neural artificial amplamente utilizada pela comunidade científica. Ela foi concebida para a resolução de problemas linearmente inseparáveis, que não poderiam ser solucionados pela rede Perceptron de Camada Única (HAYKIN, 2009). As principais características de uma MLP são (HAYKIN, 2009):

- Possui uma camada de neurônios não-computacionais, que formam a camada de entrada da MLP, a qual recebe os dados (estímulos) do ambiente;
- Uma ou mais camadas formadas por neurônios computacionais, denominadas camadas ocultas. Os neurônios computacionais capacitam a MLP para a resolução de problemas complexos, extraindo progressivamente as características mais significativas dos padrões a ela apresentados;
- Uma camada de saída, também formada por neurônios computacionais;
- Cada neurônio computacional inclui uma função de ativação não-linear, mas esta não-linearidade é suave, pois a função deve ser diferenciável em qualquer ponto. Geralmente, usa-se funções hiperbólicas.
- Há um alto grau de conectividade entre os neurônios que formam a MLP. Os neurônios pertencentes a uma camada estão ligados a todos os neurônios da camada imediatamente anterior. As ligações entre os neurônios são denominadas pesos, ou pesos sinápticos. O conhecimento adquirido pela rede neural durante o treinamento fica armazenado nos valores dos pesos.

Devido a sua robustez, as MLPs são também aplicadas em fusão de dados. Por exemplo, no trabalho de YOUSUF e KADRI (2019), é apresentada uma metodologia robusta para localização de robôs em ambientes internos e externos, por fusão de dados. A principal contribuição deste trabalho é empregar uma rede neural perceptron multicamadas para fornecer estimativas de posição de um robô, em um ambiente interno onde os sinais de GPS são bloqueados.

No trabalho de CHEN et al. (2020), é proposta a fusão de dados inerciais e visuais multifrequenciais para estimativa de atitude. A estratégia proposta é baseada em *weighted linear regression* (LWLR), MLP e *cubature Kalman filter* (CKF). Já no trabalho de MORSHEDIZADEH et al. (2018), é investigado o problema da previsão precisa da produção de energia de uma turbina eólica. Neste trabalho, é sugerida uma nova técnica de fusão de dados. Esta técnica é aplicada, combinando MLP e redes de sistema de inferência neuro-difusa adaptativa para prever o desempenho futuro de turbinas eólicas.

Outro exemplo de aplicação de MLPs em fusão de dados, é o trabalho de SENANAYAKA et al. (2019). Este trabalho investiga a detecção e o isolamento de falhas simples e mistas em uma caixa de engrenagens. Os autores propõem um algoritmo de aprendizagem híbrido, composto por classificadores baseados em perceptron multicamadas e rede neural convolucional, para diagnóstico de falhas mistas em caixas de engrenagens.

No trabalho de PUTTINAOVARAT e HORKAEW (2020), é investigado o problema de previsão e aviso antecipado de inundações. Os autores propõem um novo sistema de previsão de enchentes, baseado na fusão de *big data* meteorológicos, hidrológicos, geoespaciais e *crowdsourcing* em uma estrutura de aprendizado de máquina adaptável. Neste trabalho, foi revelado por experimentos de *benchmarking*, que o sistema configurado com uma MLP deu a previsão mais eficaz.

Os trabalhos supracitados mostram que a aplicação de MLPs em fusão de dados, continua com relevância. Isto foi um dos fatores de escolha para MLP, nesta Tese. As MLPs também são usadas, dentre as técnicas disponíveis, para navegação autônoma de VANTs. Em particular, um atrativo para usar MLPs é sua construção em *hardware* dedicado com FPGA (SAMBATTI et al., 2016; BRAGA, 2018; VITOR et al., 2020).

Esta Tese, foi inspirada pelos trabalhos de CONTE e DOHERTY (2009), SILVA (2016) e BRAGA (2018). Dado os resultados obtidos por esses trabalhos, que investigaram a navegação autônoma de VANTs, partiu-se para investigar a aplicação de MLP como fusor de dados. Em particular, a MLP autoconfigurada e construída em *hardware* dedicado com FPGA.



### 3 FUNDAMENTAÇÃO TEÓRICA

No Capítulo 2, mencionou-se várias aplicações para VANTs. Diante de vários cenários possíveis, fica claro a dificuldade de ter uma única solução para navegação autônoma. Pode-se dizer que tampouco é possível solucionar os problemas da navegação autônoma com apenas uma técnica, sendo necessário a integração de mais de uma tecnologia.

Dado que é possível aplicar várias técnicas, ou seja, há várias abordagens em estudo para possibilitar a navegação autônoma de VANTs, torna-se importante identificar quais técnicas, componentes e sensores foram usados nesta Tese.

#### 3.1 Sensores GNSS e INS

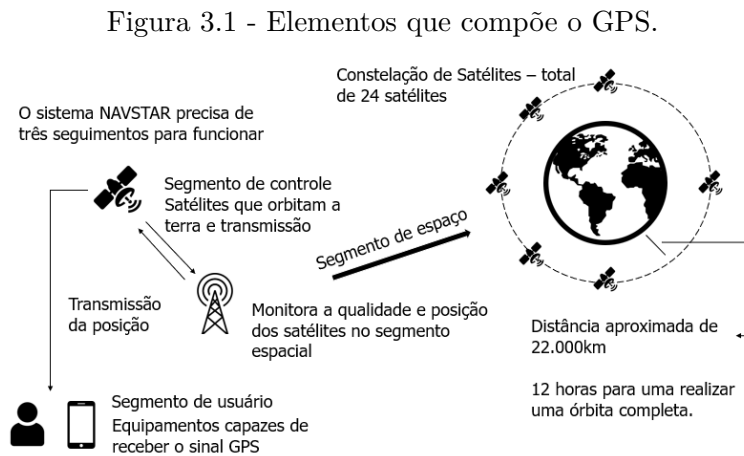
No Capítulo 1 e no Capítulo 2, mencionou-se sobre o uso de GNSS, em particular do sistema norte americano GPS, que é um sistema de posicionamento baseado em uma rede de satélites que continuamente transmite informações codificada (HOFMANN-WELLENHOF et al., 2007; GROVES, 2015; MOHAMMAD, 2020).

O GPS é, na verdade, chamado de NAVSTAR (*Navigational Satellite Timing And Ranging*). Este sistema é financiado e controlado pelo Departamento de Defesa dos Estados Unidos da América (GROVES, 2015; MOHAMMAD, 2020). O sistema NAVSTAR não é o único em operação, também existe o GLONASS, que é um sistema da Rússia. O GLONASS foi, no início da tecnologia de posicionamento por satélite, o único sistema que concorria com o GPS. No final da década de 90, os europeus criaram o seu próprio sistema de navegação por satélite, denominado de Galileo. Em 2006, a China anunciou seu próprio sistema de navegação por satélite. Este sistema é denominado de Beidou/Compass (MOHAMMAD, 2020).

A busca, dos países desenvolvidos, por um sistema de navegação por satélite global é inegável. Mas, infelizmente, o Brasil não está entre esses países, ao menos não atualmente (LIMA, 2018; MOHAMMAD, 2020). Desta forma, destaca-se que, no Brasil, o GPS é o mais usado. Por isso, cabe exemplificar o funcionamento do GPS.

O GPS, inicialmente, consistia de uma constelação de 24 satélites. Considerando a configuração antiga com 24 satélites, pode-se informar que cada um dos satélites orbita a Terra, sendo 12 em cada hemisfério. A distância dos satélites em relação a terra é próxima a 22.000km e os satélites fazem uma órbita completa a cada 12 horas. As informações transmitidas pelos satélites podem ser interpretadas pelos receptores para identificar com precisão os locais na Terra (MOHAMMAD, 2020). A

Figura 3.1 ilustra o funcionamento do GPS.



Fonte: Produção do autor.

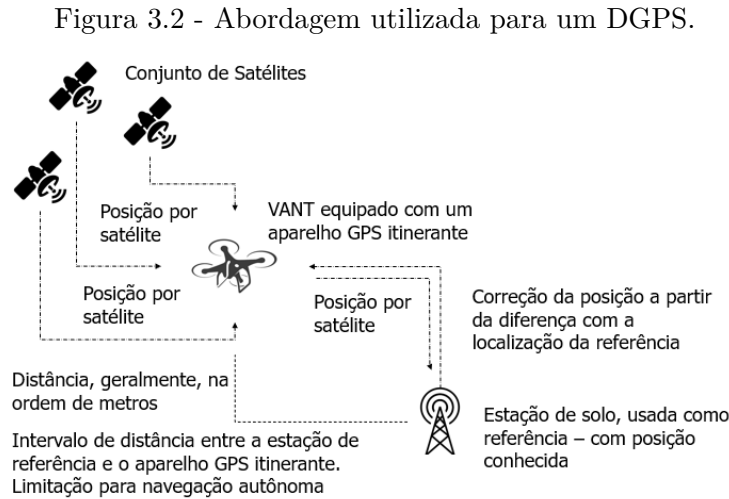
Na Figura 3.1, são ilustrados três elementos para o funcionamento do GPS. Os elementos são o segmento de espaço, segmento de controle e o segmento de usuário. Cada um deles é um dos vértices da comunicação. O segmento de espaço representa os satélites, enquanto o segmento de controle representa as antenas, usadas na transmissão de informações. Por fim, o segmento de usuário representa os dispositivos usados pelos consumidores.

Os satélites do GPS vêm sofrendo atualizações ao longo do tempo, incluindo o acréscimo da portadora L5, além das já transmitidas L1 e L2. O número de satélites em órbita aumentou e atualmente são 30 satélites operacionais. Com os atuais avanços do sistema americano, os receptores mais modernos poderão alcançar acurácia da ordem de 1 metro, pois a maioria dos efeitos da ionosfera poderão ser corrigidos (LIMA, 2018; MOHAMMAD, 2020).

Ainda no Capítulo 1, mencionou-se que o sistema GPS pode falhar. Uma das alternativas para evitar erros associados a transmissão do sinal, é utilizar o DGPS (*Differential Global Positioning Systems*). Nesta abordagem, uma estação de solo é utilizada como referência, em outras palavras, é utilizada uma estação de solo com a posição georeferenciada exatamente conhecida (LAWAL, 2019).

A estação de referência calcula os erros associados as informações vindas dos satélites, comparando sua posição com as posições medidas pelos satélites. A estação de

referência calcula a diferença entre os intervalos medidos por um aparelho GPS itinerante e a posição da estação de referência. Assim, é possível corrigir o valor obtido pelo aparelho - essa diferença calculada é chamada de correção diferencial (LAWAL, 2019). A Figura 3.2 ilustra a abordagem com um DGPS.



Fonte: Produção do autor.

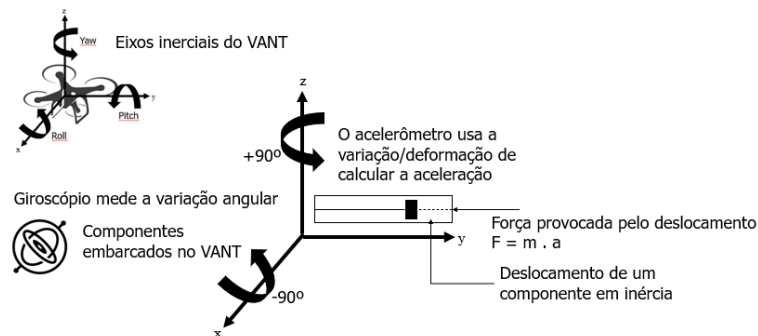
Para navegação autônoma, a abordagem que usa um DGPS possui a limitação de ter uma estação de solo, que deve ficar próximo ao aparelho GPS itinerante, como ilustra a Figura 3.2. Contudo, nesta Tese, um DGPS foi usado como referência para qualificar as estimativas feitas pela fusão de dados. Mais precisamente, utilizou-se um GPS do tipo RTK (*Real-time Kinematic*) (LAWAL, 2019).

Outro sensor usado para navegação de VANTs é o INS, que mede os movimentos lineares e angulares que ocorrem durante o voo. O INS pode ser entendido como uma combinação de outros sensores, um conjunto onde são utilizados acelerômetros e giroscópios (MOHAMMAD, 2020).

Os giroscópios medem as variações de atitude ou a taxa de rotação em relação a um espaço inercial. Na prática, os acelerômetros não são capazes de separar a aceleração total do objeto da aceleração devido à presença do campo gravitacional. Assim, as medidas fornecidas pelos acelerômetros, quando próximas a um corpo de grande massa, como a Terra, precisam ser combinadas com o conhecimento do campo gravitacional para se determinar a aceleração do objeto com respeito ao espaço inercial (MOHAMMAD, 2020). Na Figura 3.3, é ilustrado o INS e o seu funcionamento

com acelerômetros e giroscópios.

Figura 3.3 - Sistema inercial com acelerômetros e giroscópios.



Fonte: Produção do autor.

Assim, a navegação por INS combina as medições da rotação do VANT e a força específica, sendo possível calcular as estimativas da orientação, velocidade e posição em relação a um sistema de referência. Os sensores inerciais, juntamente com o modelo de gravidade e as condições iniciais do VANT, constituem a entrada de um sistema de navegação inercial (OPANASIUK, 2018).

Os acelerômetros, usualmente, medem a força específica ao longo de um único eixo e os giroscópios, normalmente, medem a velocidade angular em relação a um único eixo. Por isso, geralmente, estes componentes são montados em grupos de três, um para cada eixo e suas informações são agrupadas na IMU, embarcada no VANT (OPANASIUK, 2018).

O desempenho e o custo dos sensores inerciais variam com a tecnologia empregada. Normalmente, os sensores de alto desempenho são maiores, mais pesados e mais caros, o que pode impactar na viabilidade quando se pretende usar VANTs de baixo custo (MOHAMMAD, 2020). Por outro lado, para sensores mais baratos e com menor qualidade, os erros provenientes dos giroscópios podem afetar, diretamente, a obtenção da atitude do VANT (OPANASIUK, 2018). Além disso, também afeta a redução ou a eliminação dos efeitos da aceleração da gravidade nas medições do acelerômetro (OPANASIUK, 2018). Assim, pode-se concluir que os erros nos acelerômetros afetam diretamente a estimativa de posição da aeronave, quando a navegação é feita por um sistema inercial (MOHAMMAD, 2020).

O GPS, assim como os demais sistemas de navegação por satélite, fornece informa-

ções de localização e tempo. Este sistema, atualmente, é utilizado por aplicativos críticos para usuários militares, civis e comerciais em todo o mundo, sendo acessível a qualquer operador, usando um único receptor de GNSS (MOHAMMAD, 2020). Por isso, uma série de aplicativos que são desenvolvidos e apoiados, principalmente por esta tecnologia, tornam-se altamente dependentes deste sistema de navegação.

Grande parte dos aplicativos que usam o GPS, buscam uma solução de infraestrutura barata e altamente precisa. Contudo, uma pequena parte apresenta algum tipo de contramedida aos ataques eletromagnéticos, que o sinal GPS pode sofrer, mostrando um alto nível de vulnerabilidade aos ataques intencionais (KIM et al., 2012; KIM et al., 2013; BAO et al., 2017).

Na literatura, há fontes que demonstram os fundamentos que regem a operação do GPS (MOHAMMAD, 2020). Inclusive, as fontes abordam que é relativamente simples interferir nos sinais do GPS e, por sua vez, na localização de receptores militares ou civis (ELEZI et al., 2019). A eficácia da interferência que depende da distância e da potência do equipamento, também é de cálculo bastante simples, tornando o dimensionamento de um sistema de interferência algo acessível para pessoas ou organizações perigosas (BAO et al., 2017; JIN et al., 2019; MEI et al., 2019; ARTHUR, 2019).

Outro ponto é a construção de equipamentos interferentes. Apesar de serem ilegais em diversos países, projetos que ensinam a construir este tipo de equipamento são facilmente encontrados na rede mundial de computadores (FARIA et al., 2016). Portanto, é indiscutível a vulnerabilidade de dispositivos dependentes do GPS.

### **3.2 Navegação de VANT por imagens**

Dado os problemas com os sensores inerciais, torna-se importante buscar alternativas para navegação de VANTs, principalmente a navegação autônoma. Dentre as alternativas que se destacam na literatura, há propostas de navegação por processamento de imagens (AL-KAFF et al., 2018).

O funcionamento da navegação de VANT, por imagens, depende do embarque de um ou mais sensores imageadores no VANT. Estes sensores podem ser passivos, como, por exemplo, uma câmera (SILVA, 2016). Eles também podem ser ativos, como, por exemplo, um sensor LiDAR (BRAGA, 2018). Geralmente, os algoritmos de localização por imagem, aplicam uma correspondência entre as imagens capturadas pelo VANT e uma outra imagem georeferenciada, previamente embarcada num computador de

bordo (SILVA FILHO, 2016). Nesta Tese, utilizou-se um sensor imageador passivo, especificamente um sensor óptico.

Dentre as propostas de navegação de VANT por imagens, destacam-se Odometria Visual, *Simultaneous Localization and Mapping* (SLAM), Casamento de Padrões e Reconhecimento de marcos (AL-KAFF et al., 2018).

A Odometria Visual, também chamada de Fluxo Óptico, utiliza a sequência de imagens obtidas pelo VANT por uma câmera embarcada, durante o voo. A área de sobreposição entre essas imagens é utilizada para calcular o deslocamento da aeronave. Esta técnica busca por pontos correspondentes, em duas imagens sequenciais, que serão usados para estimar o deslocamento do movimento do VANT (AL-KAFF et al., 2018).

O processo de correspondência, na Odometria Visual, fornece o deslocamento do VANT, calculado em pixels. Contudo, geralmente, o sistema de controle de voo de um VANT precisa que este deslocamento seja fornecido em metros, isto é necessário para estimar a nova posição da aeronave. Portanto, torna-se imperativo conhecer, previamente, a resolução das imagens utilizadas e isto pode ser obtido com informações da câmera, combinadas com a altura em que a aeronave está (AL-KAFF et al., 2018).

Pode-se destacar, como vantagem da Odometria Visual, o fato da técnica não necessitar de informações prévias sobre a área sobrevoada. Além disso, a técnica tem uma alta taxa de repetibilidade, de modo que sua aplicação pode corrigir o INS com frequência (SILVA FILHO, 2016). Porém, quando utilizada por um longo período, tende a acumular erros na direção do movimento e, portanto, na posição estimada do VANT (ROOS, 2018). Por causa desta particularidade, grande parte dos estudos utilizam a Odometria Visual como uma técnica a compor, o sistema que fornece a estimativa da posição do VANT.

O Casamento de Padrões consiste em uma técnica baseada no método de registro de imagens (BRAGA et al., 2019). Para esta técnica, a estimativa da posição é calculada a partir da sobreposição de uma imagem capturada durante o voo e uma imagem previamente embarcada no VANT. Portanto, para realizar esta técnica, a aeronave deve ter a bordo uma imagem, que representa padrões, ou seja formas, de toda a área da rota sobrevoada (AL-KAFF et al., 2018).

O Casamento de padrões é considerado um método de alta precisão para navegação

por imagens (CONTE; DOHERTY, 2009). Contudo, sua principal desvantagem é necessitar de informações prévias da área sobrevoada, algo que não ocorre na aplicação da Odometria Visual.

Já o SLAM é um técnica mais ousada, pois consiste em localizar o VANT sobre uma área enquanto ela está sendo mapeada, ou seja, a localização do VANT é estimada simultaneamente com o mapeamento da região sobrevoada (AL-KAFF et al., 2018). Pela sua própria natureza, torna-se claro que o SLAM é um problema computacional complexo e com vasto campo de estudo na robótica.

A abordagem aplicada ao SLAM consiste em mapear a área sobrevoada e armazenar uma representação interna da região, em tempo real (WEI et al., 2018). Geralmente, este armazenamento acontece na forma de pontos de referência. Esta representação é então usada para fins de localização (RIZK et al., 2020).

Particularmente, o SLAM possui vantagens sobre Casamento de Padrões, porque não precisa de informações prévias da área sobrevoada. Também possui vantagens sobre a Odometria Visual, porque armazena um banco de dados com informações da área sobrevoada. Ressalta-se que isto pode ser usado para corrigir a estimativa da posição, ainda que a técnica seja aplicada por um longo período (SILVA FILHO, 2016). A desvantagem relativa a esta técnica acontece no período de mapeamento da área sobrevoada à medida que o VANT necessita visitar espaços para melhorar o mapeamento da área sobrevoada. Este processo pode impactar na autonomia do VANT, pois consome tempo de voo e energia (SILVA FILHO, 2016).

Em relação ao Reconhecimento de Marcos, pode-se dizer que, do mesmo modo que o Casamento de Padrões, esta técnica necessita de informações prévias sobre a área sobrevoada (AL-KAFF et al., 2018). O objetivo do Reconhecimento de Marcos é encontrar/reconhecer pontos de referência capturados pelo sistema de visão do VANT, durante o voo. Estes pontos de referência irão auxiliar a navegação, ajudando a corrigir o sistema de posicionamento da aeronave (SILVA FILHO, 2016).

Uma vantagem do Reconhecimento de Marcos, em relação ao Casamento de Padrões, é que esta técnica não necessita de informações prévias sobre todo a área de voo, apenas informações sobre os pontos de referência (AL-KAFF et al., 2018).

### **3.2.1 Estimação da posição do VANT por correlação de imagens**

Como informado anteriormente, há diferentes formas de estimar a posição de um VANT por processamento de imagens e uma parte desses métodos utiliza o conceito

de registro automático de imagens. Destaca-se, na literatura, duas formas de aplicar o registro automático de imagens para navegação autônoma de VANT, elas são registradas com base na correlação de imagens e o registro de imagens baseado no casamento de padrões (CONTE; DOHERTY, 2008; BRAGA et al., 2019).

A principal diferença entre as abordagens de registro baseado na correlação e no casamento de padrões consiste na forma de registro. O primeiro é baseado na correlação, que faz a convolução entre duas imagens, sendo adotado um critério de similaridade para determinar qual a melhor sobreposição entre elas. Já o registro baseado no casamento de padrões localiza um conjunto de características representativas em ambas as imagens e a correspondência é obtida pelos elementos deste conjunto (MANOJ, 2020). Em particular, nesta Tese, utilizou-se o registro baseado na correlação.

Cada uma das abordagens de registro automático de imagens possui vantagens e desvantagens. Por exemplo, os métodos baseados no casamento de padrões não operam diretamente sobre o valor de intensidade do pixel. Os padrões são, normalmente, uma informação representada por um modelo geométrico. Portanto, esta abordagem é consonante a situações onde há pontos de referência dissemelhantes no solo e que não são afetados por mudanças estacionais (por exemplo, estradas e grandes construções). Contudo, os métodos baseados no casamento de padrões podem ser apenas aplicados em locais onde há pontos de referência distintos e sua performance, geralmente, é prejudicada pelo processo de localização dos pontos de interesse, descrição e correspondência entre esses pontos. Desta forma, a abordagem pode não ser aderente para aplicações de tempo real (CONTE; DOHERTY, 2008).

Já os métodos baseados na correlação entre imagens podem ser aplicados em áreas onde não há pontos de referência no solo. Geralmente, eles são menos custosos, computacionalmente, do que os métodos de registro por casamento de padrões, sendo mais aderentes a aplicações de tempo real. Contudo, a desvantagem desta abordagem acontece quando a correção é feita entre a imagem obtida pelo VANT e uma imagem georeferenciada com cobertura de área muito maior. O fato da imagem georeferenciada cobrir um terreno maior pode provocar índices de correlação altos em lugares muito afastados da posição real da aeronave (BRAGA et al., 2019).

Existe mais de um método para fazer o cálculo da correlação entre duas imagens. Dentre os métodos, destacam-se o cálculo da correlação por soma da diferença quadrática, o cálculo da correlação cruzada normalizada e o cálculo da correlação cruzada (HISHAM et al., 2015). Nesta Tese, utilizou-se o cálculo da correlação cruzada,



porque há vários exemplos, na literatura, de sua aplicação com resultados promi-  
tentes (CONTE; DOHERTY, 2009; BRAGA et al., 2015; SILVA et al., 2015; BRAGA et al.,  
2016; BRAGA et al., 2019; SARAIVA et al., 2019; VITOR TORRES et al., 2020). Desta  
forma, cabe informar que o método proposto, nesta Tese de doutorado, para realizar  
o registro automático de imagens é baseado no algoritmo desenvolvido por CONTE  
e DOHERTY (2009).

No contexto da navegação autônoma de VANT, a correlação cruzada é calculada  
entre a imagem aérea capturada pelo VANT e a imagem georreferenciada da região  
sobrevoadada. O resultado da correlação é uma matriz cujo o seu maior valor repre-  
senta a posição do VANT na imagem georreferenciada (BRAGA et al., 2019). Isto é  
demonstrado no trabalho de VITOR TORRES et al. (2020), onde também é usada  
correlação de imagens segmentadas.

Pode-se considerar as imagens usadas no cálculo da correlação cruzada como matri-  
zes. Desta forma, o cálculo da correlação cruzada entre uma matriz  $I$  de dimensões  
 $M \times N$ , que representa a imagem do VANT, capturada durante o voo, e uma ma-  
triz  $T$  de dimensões  $P \times Q$ , que representa a imagem georreferenciada, embarcada  
previamente no computador de bordo, é determinado por (AL-KAFF et al., 2018):

$$C(k, l) = \sum_m^{M-1} \sum_n^{N-1} (I(m, n) \times \hat{T}(m - k, n - l)) \quad (3.1)$$

onde  $C(k, l)$  é a matriz de correlação  $C$  de índices  $k$  e  $l$  com  $-(P - 1) \leq k \leq (M - 1)$   
e  $-(Q - 1) \leq l \leq (N - 1)$ , a matriz  $\hat{T}$  é o complexo conjugado da matriz  $T$ .

Uma vez que a imagem capturada pelo VANT é menor do que a imagem georefe-  
renciada, a correlação cruzada entre as imagens deve ser calculada pixel por pixel,  
como se a imagem menor fosse uma janela deslizante sobre a imagem maior. Este  
cálculo é realizado entre o pixel central da imagem do VANT, com cada pixel da  
imagem georreferenciada, que é percorrida da esquerda para a direita e de cima para  
baixo. A Figura 3.4 ilustra este processo.

O maior valor na matriz  $C$ , que é o maior valor da correlação cruzada, corresponde  
ao pixel central da imagem do VANT. Considerando que a aeronave ocupa o pixel  
central da imagem capturada, por seu sensor imageador, o maior valor da matriz de  
correlação corresponde a posição da aeronave.

Quando a correlação cruzada é feita com uma imagem georeferenciada, as coordena-

Figura 3.4 - Correlação cruzada entre a imagem do VANT e a imagem georeferenciada.



Fonte: Adaptado de BRAGA (2018).

das geográficas do ponto de maior valor de correlação podem ser recuperadas. Elas utilizam as coordenadas deste ponto na imagem georeferenciada, que coincide com o ponto central da imagem do VANT. Contudo, cabe informar que antes do cálculo da correlação cruzada é necessário aplicar transformações nas imagens. Estas transformações consistem em rotação, conversão para tons de cinza, filtragem e extração de bordas (BRAGA et al., 2019).

### 3.2.2 Conversão de imagens para tons de cinza

O resultado da correlação cruzada, entre duas imagens, pode ser afetado por características nas imagens, como ângulo de captura, cor e iluminação, entre outros fatores (BRAGA et al., 2019). Estes fatores podem influenciar negativamente no resultado do processamento dos algoritmos de localização de VANT por imagens (AL-KAFF et al., 2018). Uma das formas de minimizar problemas é remover algumas características, como as cores. Uma solução comum, para remoção das cores é conversão para tons de cinza (FOLEY et al., 2013).

A conversão para tons de cinza consiste na transformação de uma imagem multibanda em uma imagem monocromática, ou seja com apenas uma cor, mantendo as características principais presentes na imagem, como as formas dos objetos nela presentes (FOLEY et al., 2013). Pode-se entender uma imagem multibanda como uma composição de duas ou mais imagens monocromáticas, que são chamadas de bandas.

Para uma imagem monocromática, em tons de cinza, cada pixel representa apenas uma quantidade de luz. Assim, ele contém apenas informações de intensidade, em que o valor é denominado nível de cinza (FOLEY et al., 2013).

O processo de conversão de uma imagem multibanda para uma imagem monocromática, em tons de cinza, permite aumento no desempenho dos algoritmos de processamento de imagens, como, por exemplo, nos algoritmos de filtragem e extração de bordas. Isto acontece porque, a redução de bandas diminui o esforço computacional desses algoritmos, uma vez que seu processamento ocorre em apenas uma banda. Geralmente, esse aumento de desempenho ocorre sem prejudicar o resultado final do algoritmo, dado que são mantidas as características principais da imagem (FOLEY et al., 2013).

O valor, ou seja, a intensidade, do pixel em uma imagem monocromática, pode ser obtido por diferentes métodos, entre eles, pela média aritmética dos valores dos pixels em cada banda da imagem multibanda. Este processo é definido por (PEDRINI; SCHWARTZ, 2007):

$$I(x, y) = \sum_n^N \frac{f(x, y)_n}{N} \quad (3.2)$$

onde  $f(x, y)_n$  é o valor, ou seja intensidade, do pixel na posição  $(x, y)$  da banda  $n$ .  $N$  é quantidade de bandas na imagem multibanda e  $I(x, y)$  o valor, do pixel, na posição  $(x, y)$ , na imagem monocromática.

Geralmente, uma imagem multibanda, obtida a partir de um sensor óptico, é composta por 3 bandas, denominadas de  $R, G$  e  $B$ , respectivamente, vermelho, verde e azul em inglês.

Para este tipo de imagem, uma outra forma de aplicar a conversão para tons de cinza, consiste nos seguintes passos; primeiro, obtém-se o valor do pixel, para cada uma das 3 bandas; depois adiciona-se, aproximadamente, 30% do valor da intensidade do pixel na banda  $R$ , mais 59% do valor da intensidade do pixel na banda  $G$  e mais 11% do valor da intensidade do pixel na banda  $B$ , ao novo pixel em tons de cinza, como mostrado por (PEDRINI; SCHWARTZ, 2007):

$$I(x, y) = 0,2989 \times R(x, y) + 0,5870 \times G(x, y) + 0,1140 \times B(x, y) \quad (3.3)$$

onde  $I(x, y)$  é o valor da intensidade do pixel, na posição  $x, y$ , na imagem em tons de cinza,  $R(x, y)$ ,  $G(x, y)$  e  $B(x, y)$  são os valores da intensidade do pixel, na posição  $x, y$ , nas bandas vermelho, verde e azul, respectivamente.

Cabe ressaltar que o processo de conversão de uma imagem multibanda para uma imagem em tons de cinza deve ser aplicado anteriormente aos processos de filtragem, extração de bordas e detecção e descrição dos pontos característicos (MANOJ, 2020).

### 3.2.3 Filtragem em imagens

Eliminar características que podem influenciar negativamente o resultado da correlação cruzada é um primeiro passo a ser aplicado sobre as imagens utilizadas para navegação autônoma, por isso é feito a conversão das imagens em tons de cinza. Contudo, a conversão em tons de cinza não elimina as características principais das imagens, isso inclui, também, os ruídos (FOLEY et al., 2013). Portanto, um outro passo para melhorar a qualidade das imagens pode ser aplicado. Ele consiste na aplicação de alguma operação de filtragem.

De modo geral, as operações de filtragem em imagens digitais são técnicas utilizadas para melhorar a qualidade de uma imagem. Estas técnicas permitem diminuir a presença de ruídos, que podem ser causados na aquisição ou nas condições nas quais a imagens são capturadas (FOLEY et al., 2013).

Cabe ressaltar que a filtragem de imagens digitais pode ser realizada no domínio da frequência ou no domínio espacial (PEDRINI; SCHWARTZ, 2007). A diferença entre as filtrações para cada tipo de domínio pode ser entendida, de modo simplificado, como: a filtragem no domínio da frequência opera sobre a transformada de Fourier da imagem e a filtragem no domínio espacial opera diretamente sobre os valores de intensidade dos *pixels* (PEDRINI; SCHWARTZ, 2007).

Com relação as técnicas de filtragem, no domínio espacial, pode-se entender que elas realizam transformações, em cada pixel, na imagem. Estas técnicas não consideram apenas o valor da intensidade de um determinado pixel, mas também, o valor da intensidade dos pixels vizinhos. O processo de filtragem é realizado com uso de matrizes, chamadas de máscaras. Estas máscaras são aplicadas sobre a imagem por uma convolução (FOLEY et al., 2013).

Há uma variedade de filtros espaciais (PEDRINI; SCHWARTZ, 2007). Contudo, para a extração de bordas, são mais interessantes filtros que realçam os detalhes da imagem, principalmente as transições entre regiões diferentes, de forma a deixá-las mais nítidas.

das, sem aumentar a intensidade dos ruídos (PEDRINI; SCHWARTZ, 2007). Por isso, usa-se estes filtros para realçar características como bordas, linhas, curvas ou sombras em uma imagem digital. Um dos filtros que se destaca na melhora de imagens digitais é o filtro Gaussiano (GEDRAITE; HADAD, 2011; JAIN; GUPTA, 2015; JULIO et al., 2015; WANG; HSIEH, 2018; BRAGA, 2018). O filtro gaussiano, também, ganha destaque como técnica de filtragem para imagens obtidas por um VANT (BRAGA, 2018).

O filtro Gaussiano é muito utilizado em processamento de imagens pela sua capacidade de suavização e remoção de ruídos na imagem (FOLEY et al., 2013). No filtro Gaussiano, os coeficientes, valores da máscara, são determinados pela função Gaussiana em duas dimensões. A função, para obter os valores da máscara é dada por (PEDRINI; SCHWARTZ, 2007):

$$G(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (3.4)$$

onde  $\sigma$  é a largura da função Gaussiana, que determina o grau de suavização do filtro, ou seja, quanto maior o valor de  $\sigma$  maior será a suavização obtida. Por conseguinte, mostra-se uma matriz  $3 \times 3$ , obtida pela Equação 3.4, quando  $\sigma^2 = \frac{2}{\pi}$ :

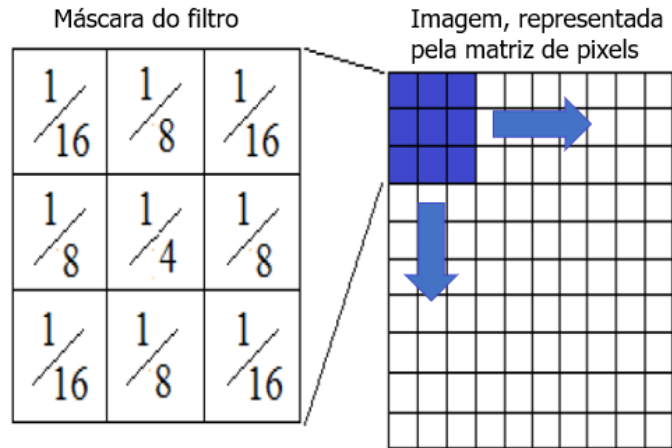
$$G(x, y) = \begin{bmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{bmatrix} \quad (3.5)$$

Considerando a máscara, mostrada na Equação 3.5, o processo de filtragem consiste em convoluir a máscara sobre toda a extensão da imagem que se deseja filtrar, como se a máscara fosse uma janela deslizante. A Figura 3.5, ilustra o processo convolução da máscara, do filtro Gaussiano, sobre uma imagem.

### 3.2.4 Extração de bordas

Imagens digitais possuem características como textura, compressão, tamanho, estatísticas (média, variância e desvio padrão), intensidade da cor dos pixels, etc (FOLEY et al., 2013). De modo geral, os algoritmos utilizados para navegação autônoma de VANT por imagens, utilizam uma ou mais características para estimar a posição do VANT sobre uma área sobrevoada (AL-KAFF et al., 2018).

Figura 3.5 - Aplicação do filtro Gaussiano.



Fonte: Adaptado de BRAGA (2018).

Contudo, a depender do tipo de algoritmo usado e do tipo de sensor imageador, algumas características não são interessantes para estimar a posição do VANT, principalmente características variantes com o ambiente (AL-KAFF et al., 2018). Desta forma, pode-se considerar que ocorrem situações em que o interesse está somente em algumas características ou partes das imagens. Existem um conjunto de características locais que podem ser compreendidas como um conjunto de padrões nas imagens, que variam de acordo com sua vizinhança. Estes padrões são identificados a partir de pequenas diferenças de uma ou mais propriedades dentro de fragmentos da imagem (FOLEY et al., 2013).

Dentre as características de uma imagem digital, que são comumente usadas em algoritmos baseados no registro automático de imagens, pode-se destacar as bordas. Isto ocorre por conta de que estes algoritmos utilizam a correspondência entre as imagens para estimar a localização do VANT, ou seja, necessidade de comparar, casar, diferentes imagens (AL-KAFF et al., 2018). Já as regiões da imagem onde se encontram bordas são mais simples de serem reconhecidas do que outros locais da imagem, além do fato de que as bordas tendem a ser invariantes (AL-KAFF et al., 2018). Por isso, o processo de extração de bordas é um dos passos aplicados nas imagens usadas para navegação autônoma, quando se usa algoritmos baseados no registro automático de imagens.

Antes do processo de extração de bordas, pode-se aplicar um outro processo, a denominada binarização da imagem. A binarização de uma imagem digital é uma

transformação, em que se atribui apenas um entre dois valores possíveis de intensidade para cada pixel (SILVA, 2016; AL-KAFF et al., 2018). Geralmente, este processo é feito após a conversão da imagem para tons de cinza e o resultado da binarização é a representação dos pixels com valores de intensidade preto ou branco, ou seja, a binarização remove a variação nos tons de cinza dos pixels.

A binarização é importante para identificar objetos e separá-los do fundo da imagem (FOLEY et al., 2013). Existem diversos métodos para realizar a binarização de uma imagem digital. Todos os algoritmos de binarização buscam por um limiar que determina o valor que minimiza a variância entre os valores de intensidade dos pixel na imagem em tons de cinza (PEDRINI; SCHWARTZ, 2007).

Em particular, nesta Tese, utilizou-se o método de Otsu (OTSU, 1979), que é um algoritmo de binarização de imagens digitais que determina o valor do limiar de forma adaptativa. A principal vantagem em determinar o limiar de forma adaptativa é o ajuste do algoritmo as variações de luminosidade no ambiente, que podem impactar na qualidade da extração de bordas. Este método foi aplicado em outros trabalhos, com bons resultados, por isso optou-se por ele nesta Tese (SILVA, 2016; BRAGA et al., 2016; BRAGA, 2018).

Após a binarização, pode-se partir para o processo de extração de bordas. A extração de bordas é um tipo de segmentação aplicada a imagens digitais (PEDRINI; SCHWARTZ, 2007). Entende-se a segmentação como um processo que tem como objetivo particionar uma imagem digital em estruturas com conteúdo semântico relevante, para uma aplicação em particular (PEDRINI; SCHWARTZ, 2007).

Como mencionado anteriormente, sobre conversão para tons de cinza, dividir uma imagem digital em partes ou em uma parte relevante reduz a quantidade de dados a serem processados. Isto também descarta informações que são consideradas menos relevantes, preservando informações essenciais sobre as formas dos objetos na imagem (KALRA; CHHOKAR, 2016).

Geralmente, técnicas aplicadas para a segmentação de imagens digitais são baseadas nas propriedades dos níveis de cinza. Estas técnicas concentram-se em descontinuidades ou similaridades (AL-KAFF et al., 2018). De modo geral, o processo de extração de bordas é realizado por uma técnica que busca por descontinuidades. Ressalta-se que estes tipos de algoritmos particionam a imagem digital com base na variação abrupta dos níveis de preto ou branco. Os pontos, linhas e bordas presentes nas imagens digitais são as descontinuidades normalmente detectadas (AL-KAFF et al.,

2018).

Dado que a extração de bordas consiste em um processo de segmentação, pode-se entender que extrair bordas é encontrar o limite entre regiões na imagem com diferentes valores de intensidade, textura ou cor. Ou seja, trata-se de encontrar na imagem digital variações rápidas ou intensas destas características.

Uma das formas comuns, para encontrar as variações rápidas nas características de uma imagem digital, como os níveis de cinza, é utilizar o gradiente da imagem. Desta forma, considerando uma imagem digital, como uma matriz de pixels,  $T(x, y)$ , onde as coordenadas  $(x, y)$  dos pixels da imagem seriam as componentes largura e comprimento, e definido a matriz  $I(x, y)$  como a intensidade dos níveis de cinza. A variação entre os valores da matriz de intensidade pode ser calculada através do gradiente, onde os locais de maior variação, ou seja maior gradiente, dentro da imagem, representam as regiões com bordas (JAIN; GUPTA, 2015).

Deste modo, o gradiente da função  $I(x, y)$  pode ser calculado a partir de suas derivadas parciais de acordo com (JAIN et al., 1995):

$$\nabla I = \begin{bmatrix} G_x \\ G_y \end{bmatrix} \quad (3.6)$$

$$\begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{DI}{Dx} \\ \frac{DI}{Dy} \end{bmatrix} \quad (3.7)$$

Onde,  $\nabla I$  representam o gradiente da matriz  $I$  da intensidade dos pixels;  $G_x$  e  $\frac{DI}{Dx}$  são as derivadas parciais da matriz  $I$  em relação a  $x$ ; e,  $G_y$  e  $\frac{DI}{Dy}$  são as derivadas parciais da matriz  $I$  em relação a  $y$ .

Além das derivadas parciais, deve-se calcular o ângulo do vetor gradiente, este fornece a direção do vetor e indica o sentido da maior variação na intensidade dos pixels da imagem. O cálculo do ângulo é definido por (JAIN et al., 1995):

$$\alpha(x, y) = \tan^{-1} \begin{bmatrix} \frac{DI}{Dx} \\ \frac{DI}{Dy} \end{bmatrix} \quad (3.8)$$

Por fim, é também necessário calcular a magnitude do vetor gradiente, pois ela



representa a intensidade da variação. Esta magnitude pode ser calculada por (JAIN et al., 1995):

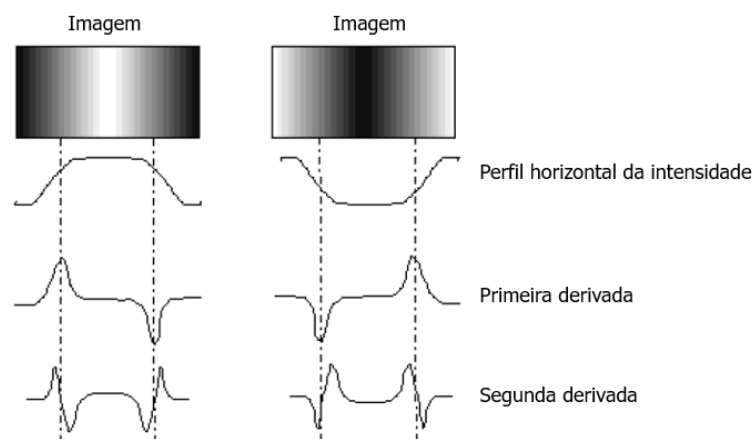
$$\|\nabla I\| = \sqrt{G_x^2 + G_y^2} \quad (3.9)$$

Além da primeira derivada, da matriz de intensidade, os algoritmos de extração de bordas, também, utilizam a segunda derivada. Neste caso a segunda derivada pode ser definida por (JAIN et al., 1995):

$$\nabla^2 I = \begin{bmatrix} \frac{D^2 I}{Dx} \\ \frac{D^2 I}{Dy} \end{bmatrix} \quad (3.10)$$

Cada derivada, da matriz de intensidade, denota um aspecto. Por exemplo, em imagens digitais, em tons de cinza, a primeira derivada do perfil horizontal é positiva nas transições de regiões escuras para claras e negativa nas transições de regiões claras para escuras. Já a segunda derivada se comporta de maneira contrária, sendo que tem valor zero na região onde o nível da intensidade de cinza é constante (PEDRINI; SCHWARTZ, 2007). A Figura 3.6 ilustra os aspectos de cada derivada da matriz intensidade.

Figura 3.6 - Aspecto das derivadas da matriz intensidade.



Fonte: Adaptado de PEDRINI e SCHWARTZ (2007).

O processo de extração de bordas também ocorre pela aplicação de uma máscara, semelhante ao processo de filtragem, com o filtro gaussiano, ou seja, por uma convo-

lução. Neste caso, o resultado da aplicação da máscara consiste numa aproximação numérica, por meio de um operador matricial. A aproximação do gradiente da matriz de intensidade pode ser feita por (PEDRINI; SCHWARTZ, 2007):

$$\nabla I = \begin{bmatrix} I[i, j + 1] - I[i, j] \\ I[i, j] - I[i + 1, j] \end{bmatrix} \quad (3.11)$$

Na Equação 3.11, o gradiente é aproximado a partir da diferença entre dois pixels vizinhos e este conceito é aplicado em muitos algoritmos para extração de bordas, sendo que cada um utiliza diferentes aproximações do gradiente, resultando em diferentes operadores. Contudo, destaca-se dois operadores clássicos chamados de Sobel e Canny (JAIN et al., 1995).

O operador Sobel realiza a detecção de bordas com base no cálculo da magnitude do vetor gradiente de um pixel em relação aos seus vizinhos (PEDRINI; SCHWARTZ, 2007). O operador Canny é um algoritmo que busca otimizar a localização de bordas na presença de ruído (JUMAAT et al., 2014; KUTTY et al., 2014). O operador Canny realiza uma convolução de um filtro gaussiano na imagem, suaviza o ruído presente e localiza as bordas calculando a magnitude do gradiente (KALRA; CHHOKAR, 2016). Os operadores Canny e Sobel são considerados extratores de bordas clássicos, pois já foram bastante aplicados (PEDRINI; SCHWARTZ, 2007; KALRA; CHHOKAR, 2016; BRAGA, 2018).

### 3.2.5 Extrator de bordas neural

Um outro operador para extração de bordas, que também é aplicado ao estudo da navegação autônoma de VANT por imagens, é a rede neural do tipo MLP (SILVA et al., 2015; SILVA, 2016; BRAGA, 2018).

A MLP consiste em um modelo matemático, que pode ser utilizado como filtro para a extração de borda. Isto acontece porque as MLPs são naturalmente um filtro não-linear, devido a não-linearidade da função de ativação dos neurônios computacionais e dos pesos sinápticos (PATINO-ESCARCINA; COSTA, 2007; HAYKIN, 2009).

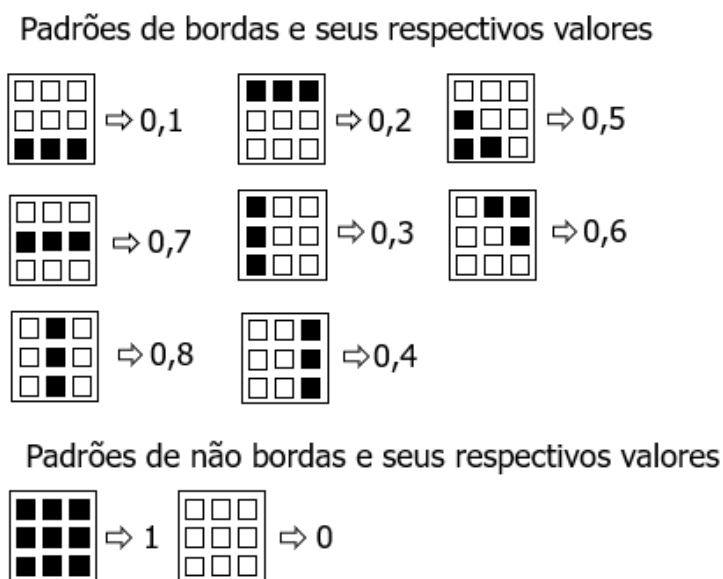
Um dos primeiros trabalhos que investigou o uso de redes neurais como extrator de bordas foi o de SRINIVASAN et al. (1994). Neste trabalho, descreveu-se uma rede neural sensível a bordas, em um campo receptivo largo, assumindo que na retina as bordas são detectadas por camadas neuronais iniciais. Além da detecção de bordas, também foi um dos primeiros experimentos que mostrou o uso de redes neurais para

compressão de uma imagem digital.

Atualmente, a MLP tem sido empregada com sucesso no processo de detecção de bordas, principalmente devido à sua capacidade de generalização, tolerância a variações nas imagens e nos ruídos, desde que com uma arquitetura bem definida e devidamente treinada (SILVA, 2016; REVELO et al., 2019).

Nesta Tese de doutorado, utilizou-se a técnica mostrada no trabalho de BRAGA (2018). Esta técnica consiste em treinar uma MLP para reconhecer padrões de bordas e padrões de não bordas (ausência de bordas). Para o treinamento da MLP, é necessário, primeiro, definir um conjunto de matrizes,  $3 \times 3$ , que denotam os padrões de bordas e não bordas, para uma imagem binarizada, ou seja, uma imagem que passou pelos processos de transformação para tons de cinza e binarização, descritos anteriormente. Cada uma dessas matrizes, representa uma janela de pixels e cada matriz é associada a um valor de ativação. A Figura 3.7, ilustra padrões de bordas para treinamento do extrator de bordas neural. Cabe ressaltar que existem mais padrões de bordas e que a dimensão das matrizes de pixels, também, podem ser  $5 \times 5$ ,  $7 \times 7$  ou ainda valores maiores, contudo, no contexto desta Tese, utilizou-se apenas matrizes de dimensão  $3 \times 3$ , pois esta dimensão apresentou bons resultados como mostrado no trabalho de BRAGA et al. (2019).

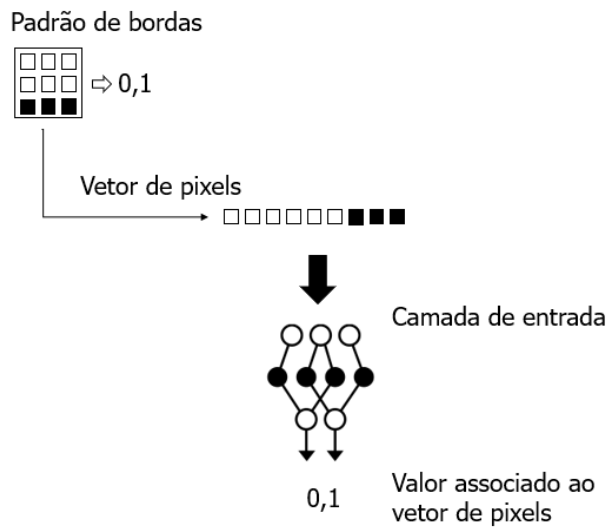
Figura 3.7 - Exemplos de padrões de bordas, para treinamento da MLP.



Fonte: Próprio autor.

A MLP é treinada para receber um vetor como entrada, que representa os valores das matrizes de pixels. Como resultado da ativação a MLP responde com o valor associado ao vetor de entrada. Por isso, é possível utilizar a MLP para identificar janelas, no tamanho de  $3 \times 3$  pixels que possuem um determinado padrão de bordas, numa imagem binarizada. A Figura 3.8, ilustra a representação de um padrão de borda no vetor de pixels, usado para ativação da MLP.

Figura 3.8 - Representação de um padrão de bordas em vetor de pixels.



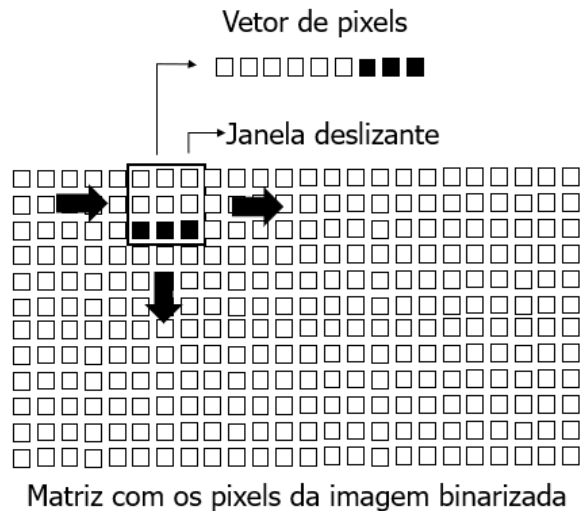
Fonte: Próprio autor.

De forma semelhante aos operadores clássicos, de extração de bordas, a ativação da MLP para a extração de bordas é feita por meio de uma janela deslizante, que percorre uma imagem binarizada, da esquerda para a direita e de cima para baixo, pixel a pixel, formando, a cada passo, o vetor de entrada com dimensão  $1 \times 9$ , para ativação da MLP. A Figura 3.9, ilustra o deslocamento da janela deslizante, para formação do vetor de pixels, utilizado na ativação da MLP.

### 3.3 Fusão de dados

A aplicação de múltiplos sensores em entidades não humanas, como máquinas e sistemas, é a principal forma de fornecer alguma inteligência para estes autômatos (LIGGINS et al., 2017). Com a introdução dos sensores, é possível permitir que máquinas possam agir e interagir em ambientes não estruturados sem ou com pouca presença humana (KHALEGHI et al., 2013).

Figura 3.9 - Deslocamento da janela deslizante para extração de bordas.



Fonte: Próprio autor.

De certo modo, pode-se dizer que os sensores permitem que um sistema aprenda o estado do ambiente, onde ele está inserido (FOURATI, 2020). Com isso, é possível ao sistema criar uma representação, um modelo do ambiente, e isto permite a interação da máquina de modo preciso, de maneira que ela passa a se comportar inteligentemente (YOUSUF; KADRI, 2019).

Nos últimos anos, a tecnologia de fabricação de sensores teve grande evolução (FOURATI, 2020). Como consequência, ocorreu o surgimento de uma diversidade de sensores e houve um decréscimo geral no custo destes dispositivos (FRADEN, 2010; FOURATI, 2020). Do ponto de vista econômico, a redução do custo de fabricação de sensores permitiu a entrada desses equipamentos em várias atividades e, por isso, é possível encontrar equipamentos inteligentes que funcionam com sensores de baixo custo (ENAMAMU et al., 2017; FOURATI, 2020).

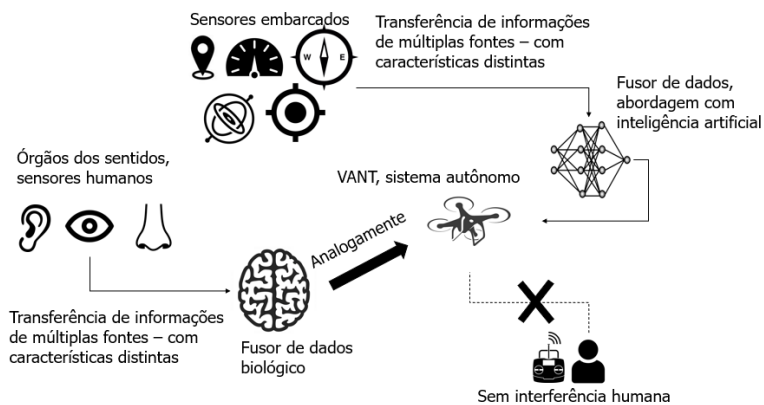
Junto da proliferação de sensores, vieram exigências como grande capacidade de armazenamento e grande velocidade de processamento, para tratar os dados provenientes das várias fontes em tempo real (ANTONY, 2015). Essas circunstâncias favoreceram o surgimento de pesquisas que investigam a combinação de dados de múltiplos sensores, esta combinação é denominada de fusão de dados (KHALEGHI et al., 2013; ANTONY, 2015).

Pode-se entender a fusão de dados, como um processo contínuo que lida com a

associação, correlação e combinação de dados e informações de várias fontes, para alcançar a posição e/ou comportamento adequado de um sistema, imerso em um ambiente (FRADEN, 2010; FOURATI, 2020). Além disso, também identifica estimativas e avaliações de situações que este sistema possa passar (ANTONY, 2015).

O objetivo da fusão de dados é transformar os dados provenientes de vários sensores em processos de inferência ou decisão (ANTONY, 2015). Do ponto de vista biológico, pode-se entender os sensores como algo análogo aos órgãos do sentido, que permitem um ser humano interagir com o ambiente no qual está inserido (FOURATI, 2020). A Figura 3.10 ilustra esta analogia com uma comparação entre o fusor de dados biológico (cérebro) e o fusor neural, que é baseado numa rede neural. Ainda, na Figura 3.10 sugere a aplicação da fusão de dados na navegação autônoma de VANT, neste caso, o fusor de dados recebe informações dos sensores embarcados na aeronave, sensores inerciais e/ou sensor imageador.

Figura 3.10 - Analogia entre sensores e os órgãos do sentido.



Fonte: Próprio autor.

Pode-se classificar a interação dos sensores na fusão de dados como competitiva, complementar, cooperativa ou ainda como uma união entre elas (NASHMAN, 1993; LIGGINS et al., 2017). Diz-se que a interação é competitiva quando sensores de um mesmo sistema estão adquirindo informações redundantes. Desta forma, as informações apresentadas pelos sensores ao fusor de dados, são iguais em características e não acrescentam qualquer conhecimento adicional para o sistema (LIGGINS et al., 2017). Geralmente, usa-se sensores competitivos para se ter um incremento na confiança de uma informação obtida do ambiente.

A interação complementar ocorre quando dois ou mais sensores estão adquirindo informações sobre um mesmo aspecto do ambiente, mas com características diferentes. Por exemplo, considere um sensor usado para definir se um sistema está em um ambiente claro e um outro sensor que captura a intensidade de luz do ambiente (LIGGINS et al., 2017).

A interação cooperativa acontece quando as informações adquiridas por dois ou mais sensores, são complementares ou dependentes. Neste sentido, pode-se pensar em um sensor que complementa ou depende da informação de um outro sensor para poder processar e entregar sua informação ao fusor de dados (LIGGINS et al., 2017). Com relação a fusão de dados, aplicada à navegação autônoma de VANT, a interação entre os sensores é uma combinação das interações competitiva, complementar e cooperativa.

Uma das particularidades da fusão de dados, além da classificação das interações dos sensores, é que ela pode ser efetuada em níveis. Entende-se a fusão em níveis quando um ou mais sensores não informam diretamente o dado capturado do ambiente, mas o resultado de algum processamento sobre esta informação (FOURATI, 2020). Isto acontece, por exemplo, quando utiliza-se as técnicas de navegação autônoma por imagens. Nestes casos, geralmente não se passa a imagem capturada pelo sensor imageador ao fusor de dados, o que se entrega ao fusor de dados é o resultado de algum processamento sobre a imagem.

A necessidade de aplicar a fusão de dados já foi bastante debatida na comunidade científica e, atualmente, é a principal solução quando se pensa em sistemas inteligentes autônomos (FOURATI, 2020). Entretanto, um dos desafios é a escolha da técnica para implementar o fusor de dados (FOURATI, 2020). Neste sentido, uma técnica já consolidada é o FK (P.; KHANDEKAR, 2019).

### 3.3.1 Filtro de Kalman

O Filtro de Kalman (FK) é uma técnica consolidada e aplicada na resolução de vários problemas (NGO et al., 2009). A técnica surgiu na década de 60 e até hoje encontram-se estudos que aproveitam esta abordagem, principalmente para navegação autônoma (LI et al., 2015; BAO et al., 2017; TAN et al., 2018; PETUKHOV et al., 2019; ZHANG; XU, 2019). O FK surgiu como uma solução recursiva para o problema da filtragem linear de dados discretos (KALMAN, 1960).

No FK, o comportamento de um sistema, em um ambiente, é caracterizado por um

conjunto de relações (expressões) lineares, em que o p̄metro que indica o estado do sistema ́ recursivamente atualizado atrav́s da aquisiç̄o de novas informaç̄es do ambiente (LI et al., 2015).

Dado que as informaç̄es tamb́m s̄o vulneráveis a erros, ́ necesśrio um modelo, tamb́m linear, para o processamento de informaç̄es. A partir deste modelo, ́ calculada uma estimativa para as informaç̄es obtidas do ambiente e se estas forem iguais às novas informaç̄es adquiridas, do ambiente, entende-se que o comportamento do sistema naquele ambiente est́ correto ou est́vel (LI et al., 2015).

O FK usa as característicás estatísticás de um sistema para, recursivamente, calcular uma estimativa do seu estado no ambiente. Desta forma, as relaç̄es lineares que caracterizam o FK dividem-se em dois grupos: relaç̄es de prediç̄o e relaç̄es de correç̄o (KALMAN, 1960). As relaç̄es de prediç̄o determinam uma estimativa à priori do estado atual do sistema e da covariância do erro, ou diferençã, entre o estado atual e o estado anterior do sistema. As relaç̄es de correç̄o s̄o aplicadas recursivamente e inserem novas informaç̄es na estimativa efetuada à priori para o estado do sistema, gerando uma estimativa à posteriori melhorada (LI et al., 2015).

Durante o processo recursivo do FK ́ calculado um fator de ganho a cada interaç̄o, ou seja, a cada mudançã de estado do sistema. Este fator de ganho serve para minimizar a covariância do erro. Tamb́m, durante o processo de estimativa à posteriori, ́ utilizado para determinar uma nova estimativa à priori.

O FK ́ interessante para a fusão de dados, porque as informaç̄es obtidas por um grupo de vários sensores podem ser combinadas usando um FK, de forma a fornecer uma estimativa do estado atual e uma prediç̄o do futuro estado de um sistema. O estado estimado pode corresponder, por exemplo, à posiç̄o atual de um VANT, juntamente com sua e velocidade e direç̄o (TAN et al., 2018).

A descriç̄o matemática do FK ́ extensa, mas ́ interessante definir seu funcionamento. Por isso, de modo sucinto, s̄o definidas em seguida as equaç̄es usadas na aplicaç̄o/construç̄o do FK - todas as equaç̄es foram derivadas da publicaç̄o de (KALMAN, 1960).

Dado que, o FK ́ uma soluç̄o para se encontrar uma estimativa do estado de um sistema e um modelo do comportamento do sistema. Pode-se definir ambos como:

$$x_{k+1} = A_k x_k + B u_k + w_k \quad (3.12)$$



$$z_k = H_k x_k + v_k \quad (3.13)$$

Onde,  $x_k$  denota o estado atual do sistema,  $A_k$  é a matriz que relaciona o estado no instante  $k$ , com o estado do instante  $k + 1$ .  $B$  é a matriz que relaciona a entrada de controle do sistema e  $u_k$  é um vetor com as informações de entrada, coletadas do ambiente.  $z_k$  é o vetor de informações, passado para o modelo do sistema,  $H_k$  é a matriz que relaciona o estado com as informações de  $z_k$  e, por fim,  $w_k$  e  $v_k$  representam ruídos associados ao processo do FK e as informações obtidas do ambiente.

Os ruídos  $w_k$  e  $v_k$  são variáveis aleatórias. E, geralmente, assume-se que estas variáveis possuem as seguintes características: são independentes uma da outra; ruído branco; e apresentam uma característica de distribuição de probabilidade normal.

O processo de computação das estimativas do estado do sistema pode ser entendido da seguinte forma, considerando  $\bar{x}_{k-1}$  como a estimativa do estado à priori, no instante  $k$ , e  $\bar{x}_k$  como a estimativa do estado à posteriori, no instante  $k$ , e também, dado as informações  $z_k$ , Pode-se então, definir uma estimativa para os erros à priori e à posteriori através das seguintes expressões:

$$e_{k-1} = x_k - \bar{x}_{k-1} \quad (3.14)$$

$$e_k = x_k - \bar{x}_k \quad (3.15)$$

Com isso, a estimativa das covariâncias dos erros são expressos por:

$$P_{k-1} = E[e_{k-1}e_{k-1}^T] \quad (3.16)$$

$$P_k = E[e_k e_k^T] \quad (3.17)$$

A dedução das relações lineares do FK inicia-se com a busca de uma expressão que descreva a estimativa do estado à posteriori, a partir da combinação linear da estimativa à priori e da diferença obtida entre a informação atual  $z_k$  e do valor  $H_k \bar{x}_{k-1}$ . Assim, a expressão que descreve a estimativa do estado à posteriori pode

ser definida como:

$$\bar{x}_k = \bar{x}_{k-1} + K(z_k - H_k \bar{x}_{k-1}) \quad (3.18)$$

A diferença  $z_k - H_k \bar{x}_{k-1}$ , denota o erro da estimação do estado do sistema, em relação as informações obtidas do ambiente e  $K$  denota um fator de ganho, que permita minimizar a covariância do erro  $P_k$ .

Com relação a minimização do erro de estimativa, substituindo a expressão da Equação 3.18 na expressão da Equação 3.15, segue com:

$$e_k = \bar{x}_k - \bar{x}_{k-1} + K(z_k - H_k \bar{x}_{k-1}) \quad (3.19)$$

Partindo da expressão da Equação 3.19, substituindo a expressão da Equação 3.17, tem-se:

$$P_k = [(\bar{x}_k - \bar{x}_{k-1} + K(z_k - H_k \bar{x}_{k-1})) \times (\bar{x}_k - \bar{x}_{k-1} + K(z_k - H_k \bar{x}_{k-1}))^T] \quad (3.20)$$

A partir da expressão da Equação 3.20, calcula-se a derivada da expressão em relação a  $K$  e iguala-se a zero. Desta forma, obtem-se uma expressão para  $K$  que minimiza a expressão da Equação 3.20. A expressão para  $K$  é definida como:

$$K_k = P_{k-1} H_k^T (H_k P_{k-1} H_k^T + R_k)^{-1} = \frac{P_{k-1} H_k^T}{H_k P_{k-1} H_k^T + R_k} \quad (3.21)$$

Na expressão da Equação 3.21,  $R_k$  representa a covariância do erro de medida entre as informações reais do ambiente e as geradas pelo modelo.

Apesar do FK ser uma técnica consolidada, ainda existem problemas associados à sua aplicação na fusão de dados, em particular na navegação autônoma de VANT. Por isso, alguns estudos buscam alternativas ao FK (KONG et al., 2019; WANG; WANG, 2019). Uma das alternativas é usar redes neurais artificiais (SAMBATTI et al., 2012; MOSBAH; EL-HAWARY, 2019; CHEN et al., 2020).

### 3.3.2 Redes neurais artificiais

As Redes Neurais Artificiais (RNA) são modelos matemáticos que representam os princípios de atividades do cérebro, com base na neurobiologia e na teoria do comportamento (MIRJALILI et al., 2020). Estes modelos são adequados para resolver problemas que envolvem classificação ou predição, usando informações conflitantes ou incompletas, tais como em aplicações de reconhecimento de padrões, otimização, controle, compressão de dados, diagnóstico e aproximação de funções, entre outras (MIRJALILI et al., 2020).

As redes neurais artificiais diferem de outros modelos computacionais por possuir uma arquitetura composta de várias unidades de processamento simples interconectadas, denominadas neurônios, formando uma estrutura semelhante a uma rede de tecido nervoso cerebral (DASGUPTA, 2020).

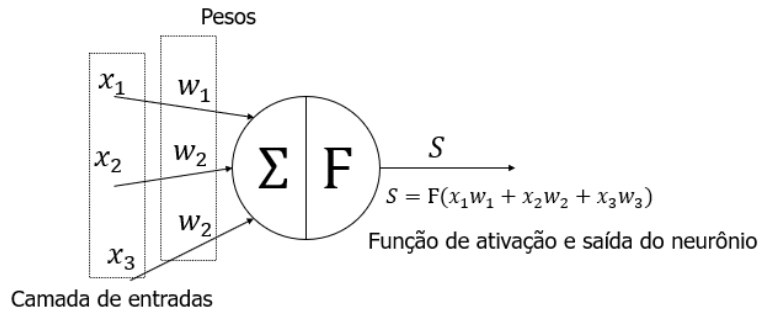
O princípio de funcionamento das redes neurais está diretamente ligado ao funcionamento de cada neurônio. Quando excitado o neurônio processa as informações apresentadas nas suas entradas (dendritos) até sua saída (axônio), segundo uma função de transferência, inspirando-se no comportamento de neurônios biológicos. Como no cérebro, as entradas da rede neural correspondem aos sensores e a saída aos neurônios motores (atuadores) que respondem aos estímulos (DASGUPTA, 2020).

O funcionamento básico de cada neurônio consiste no somatório da multiplicação de cada elemento de entrada pelo peso associado a ele. O peso é um fator reforçador ou inibidor, utilizado para ajustar a importância da entrada no estímulo do neurônio (MIRJALILI et al., 2020). O resultado do somatório passa por uma função de transferência, produzindo a saída do neurônio, que é submetida as entradas dos neurônios da camada seguinte, se esta existir. A Figura 3.11 ilustra o processo de ativação de um neurônio artificial.

O interesse em redes neurais está na possibilidade de construção de redes de computação artificiais, que têm a capacidade de imitar o cérebro nas tomadas de decisão e no raciocínio, envolvendo informações que apresentam características complexas, ruidosas, irrelevantes e/ou parciais (MELLO; PONTI, 2018). Estas características tornam redes neurais interessantes para aplicação em fusão de dados.

A forma como as redes neurais armazenam conhecimento está relacionada à aprendizagem, que é a medida da variação dos pesos no tempo. Desta forma, a memória de uma rede neural é resultado do processo de atualização dos pesos das conexões,

Figura 3.11 - Funcionamento da ativação do neurônio artificial.



Fonte: Próprio autor.

provocado pela aquisição de novos conhecimentos (MELLO; PONTI, 2018).

Para uma rede neural, a aprendizagem pode ser não-supervisionada ou supervisionada. Na primeira forma, o processo de aprendizagem é iterativo e continua até se atingir uma resposta estável para um determinado conjunto de padrões de entrada. Na segunda forma, é feito o monitoramento da amostragem do conjunto de padrões de entrada e saída definidos, observando-se o princípio da aprendizagem por condicionamento (HAYKIN, 2009).

Dada sua robustez, as redes neurais são usadas como alternativas ao FK (NOWOSAD, 2001; CINTRA; CAMPOS VELHO, 2019). Partindo deste princípio, pode-se então utilizar redes neurais como fusor de dados, em que o nome estabelecido, nesta Tese, é fusor neural. Cabe ressaltar que, o objetivo com redes neurais não é apenas substituir o FK, a abordagem precisa ser vantajosa. Desta forma, a principal vantagem está na redução da complexidade e na construção de um *hardware* de alto desempenho.

### 3.4 Desenvolvimento em FPGA

Como informado, anteriormente, pode-se usar redes neurais para substituir o FK. Em particular, pode-se usar redes do tipo MLP (CINTRA; CAMPOS VELHO, 2019). Com relação à fusão de dados, de nada adianta trocar a técnica do fusor se não ocorrer ganho, principalmente ganho de desempenho. Neste sentido, a literatura aponta como solução a construção de MLPs em *Field Programmable Gate Array* (FPGA) (ZHAI et al., 2016; GAIKWAD et al., 2019).

FPGAs são dispositivos semicondutores baseados em uma matriz de blocos lógicos configuráveis (CLBs) conectados por meio de interconexões programáveis (XILINX,

2020). CLBs são elementos para a construção e/ou configuração de lógica combinacional simples e, também, flip-flops para a realização de lógica sequencial. São unidades lógicas implementadas, utilizando blocos de memória e operações booleanas simples de alta velocidade (XILINX, 2020).

FPGAs podem ser reprogramados para a aplicação desejada ou para requisitos de funcionalidade após a fabricação. Este recurso distingue FPGAs de Circuitos Integrados Específicos de Aplicativo (ASICs), que são fabricados de maneira personalizada para tarefas específicas e, portanto, não podem ser modificados (XILINX, 2020).

Na esfera da computação, o mais comum de se encontrar é a programação feita em software. Isto porque o desenvolvimento em software fornece flexibilidade, permitindo a alteração e construção de aplicações rapidamente (PIMENTA, 2017). Contudo, esta abordagem possui um desempenho inferior, quando comparada ao desenvolvimento em *hardware*, principalmente com relação ao consumo de energia (PIMENTA, 2017). Neste sentido, o FPGA torna-se uma solução intermediária entre o desenvolvimento em ASICs e o desenvolvimento em software, uma vez que esses dispositivos, combinam benefícios de hardware e software, e podem ser configurados para se comportarem como um circuito lógico em *hardware* dedicado.

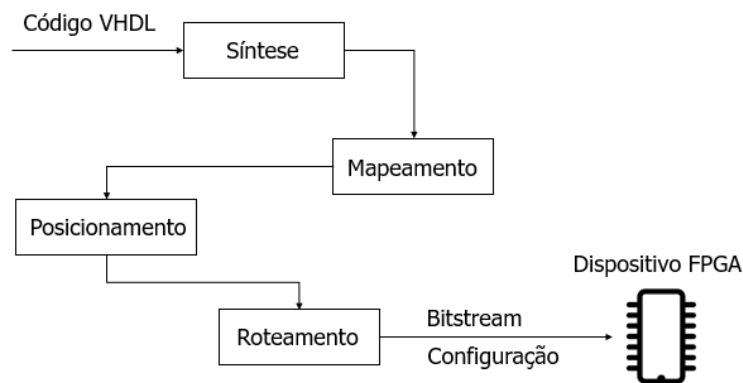
Existe mais de uma categoria de dispositivos lógicos programáveis. Algumas são *Programmable Array Logic* (PAL), *Generic Array Logic* (GAL) e *Complex Programmable Logic Device* (CPLD). Apesar das outras categorias, o FPGA se destaca pois, possui maior flexibilidade devido à abundância e ao tamanho reduzido de suas unidades básicas, o que permite a configuração de sistemas complexos com um custo computacional reduzido (PIMENTA, 2017).

Tradicionalmente, o desenvolvimento de aplicações em FPGAs é feito através de linguagens de descrição de *hardware* como C2H, System C, System Verilog e *Very High Speed Integrated Circuits Hardware Description Language* (VHDL). De certo modo, essas linguagens se aproximam mais de linguagens de desenvolvimento de software e, por isso, costumam facilitar o desenvolvimento de aplicações em FPGAs. Dentre estas linguagens, destaca-se a VHDL, que possibilita a descrição em alto nível do comportamento lógico da aplicação e do fluxo interno de dados dentro do dispositivo FPGA.

Para a construção do *hardware* em FPGA, utiliza-se um software responsável pelas etapas de configuração do dispositivo. A primeira etapa é a sintetização do código

fonte em VHDL, em que a sintetização transforma a lógica em alto nível em portas lógicas interconectadas. A próxima etapa é o processo de mapeamento, em que as portas lógicas são separadas em grupos, para serem melhor adaptadas aos recursos lógicos do dispositivo FPGA. Esta etapa é decisiva, pois nela é verificado se é possível a configuração no dispositivo, ou seja, se há blocos lógicos suficientes para construção do *hardware*. A terceira etapa é o posicionamento, em que ocorre a avaliação sobre qual bloco lógico cada grupo de portas lógicas deve ser configurado. Por fim, a última etapa é o roteamento, em que são definidas as interconexões que irão transportar os sinais entre os blocos. Ao final destas etapas, é gerado um bitstream que contém as informações necessárias para configurar o dispositivo FPGA com a aplicação projetada (PIMENTA, 2017). A Figura 3.12 ilustra o fluxo de configuração de um FPGA.

Figura 3.12 - Fluxo de construção/configuração do *hardware* em FPGA.



Fonte: Próprio autor.

Como informado, é possível que o projeto do *hardware* não caiba no FPGA. O espaço está relacionado a quantidade de CLBs que o dispositivo possui. Esta questão é um problema quando se pretende construir uma MLP com FPGA, pois a quantidade de neurônios e camadas pode esgotar os recursos programáveis no FPGA (SANAULLAH et al., 2018).

Por exemplo, nos estudos de GAIKWAD et al. (2019), os autores abordam a construção de um sistema de reconhecimento e classificação de atividades humanas. Para realizar a classificação de atividades, foi construída uma MLP com FPGA. Contudo, neste estudo foi necessário testar dez modelos MLP diferentes com topologias distintas para poder selecionar uma mais adequada ao problema e, ainda assim, não é

afirmado pelos autores que a topologia da MLP é a melhor possível ou é otimizada.

Outro exemplo é o estudo de ALILAT e YAHIAOUI (2019), os autores propõem uma codificação eficiente, que proporciona uma implementação de MLP em FPGA com uma arquitetura ótima. Porém, os próprios autores afirmam que a abordagem desenvolvida e implementada em FPGA foi avaliada no contexto da aplicação em critérios objetivos, de acordo com dois aspectos de software e hardware. Em outras palavras, não é possível garantir que a técnica entregue uma arquitetura otimizada quando é variado o problema ou o dispositivo FPGA usado.

Muitos trabalhos utilizam redes neurais do tipo MLP. Contudo, no trabalho de VITOR TORRES et al. (2020) foi aplicada uma rede neural sem pesos. Esta rede neural também foi implementada em FPGA e os autores fizeram um estudo de *benchmarking* sobre o resultado da correlação de imagens da floresta amazônica.

Diante desse problema, torna-se importante buscar uma abordagem que auxilie na construção de uma MLP com arquitetura otimizada, sem depender de algum aspecto físico do dispositivo FPGA, ou ainda, da linguagem de descrição de *hardware* disponível. Uma abordagem que se destaca e que foi utilizada nesta Tese é o Algoritmo de Colisão de Múltiplas Partículas (MPCA) (ANOCHI et al., 2015).

### 3.4.1 Rede neural autoconfigurada

Dado que o tamanho da topologia de uma MLP pode ser um problema para construção da rede com FPGA, torna-se importante utilizar uma abordagem que entregue uma topologia otimizada. Neste sentido, o MPCA é uma técnica que procura resolver esta questão como um problema de otimização (SACCO; DE OLIVEIRA, 2005).

Para muitas aplicações, tais como engenharia, atividades empresariais, na indústria, entre outras, a otimização é paradigma importante. A otimização está relacionada, geralmente, a minimização do consumo de energia e custos, ou a maximização do lucro, da produção, do desempenho e a eficiência (SINGIRESU, 2009). Desta forma, faz todo sentido pensar na implementação de uma MLP em FPGA como um problema de otimização, dado a preocupação com o consumo do espaço lógico programável.

O MPCA é um algoritmo meta-heurístico. A maioria dos algoritmos meta-heurísticos, são muitas vezes inspirados na natureza, e estão entre os algoritmos mais utilizados em problemas de otimização (ANOCHI et al., 2015). O MPCA é baseado no Algoritmo de Colisão de Partículas (PCA) e este, por sua vez, foi pensado a partir da física das reações de colisão de partículas em um reator nuclear, em

particular nos comportamentos de espalhamento e absorção (SACCO; DE OLIVEIRA, 2005). A estrutura proposta para o algoritmo MPCA, utiliza-se um conjunto de  $n$  partículas explorando, de maneira independente, porém colaborativa, o mesmo espaço.

O MPCA já foi aplicado com sucesso em outros problemas (ANOCHI et al., 2015), inclusive problemas relacionados ao navegação autônoma de VANTs (SILVA, 2016; BRAGA et al., 2019).

O objetivo do MPCA, é encontrar a arquitetura otimizada de uma MLP. Para isso, o algoritmo determina os valores ideais para os parâmetros da rede neural. Esta otimização acontece pela minimização de uma função de custo, que é definida como (CARVALHO et al., 2011):

$$J(Q) = \Phi(x, y) \times \frac{\rho_1 E_t(Q) + \rho_2 E_g(Q)}{\rho_1 + \rho_2} \quad (3.22)$$

$$\Phi(x, y) = \theta_1 [e^{x^2} + \theta_2 [y] + 1] \quad (3.23)$$

Onde,  $E_t$  e  $E_g$  são, respectivamente, os erros de treinamento e generalização;  $\rho_1$  e  $\rho_2$  são parâmetros relacionados ao equilíbrio entre o erro de treinamento e o erro de generalização;  $\Phi(x, y)$  é a medida da complexidade da rede neural, com  $x$  sendo o número de neurônios e  $y$  o número de épocas até a convergência durante a fase de treinamento;  $\theta_1$  e  $\theta_2$  são parâmetros de ajuste, por fim  $Q$  é o custo da rede neural. Os valores para os parâmetros de ajuste  $\theta_1$  e  $\theta_2$  foram obtidos da literatura, especificamente, do trabalho de SACCO e DE OLIVEIRA (2005). Com relação aos valores  $\rho_1$  e  $\rho_2$ , estes foram obtidos a partir do trabalho de ANOCHI et al. (2015). Os demais hiperparâmetros para treinamento da rede neural são obtidos de maneira automática pelo MPCA.



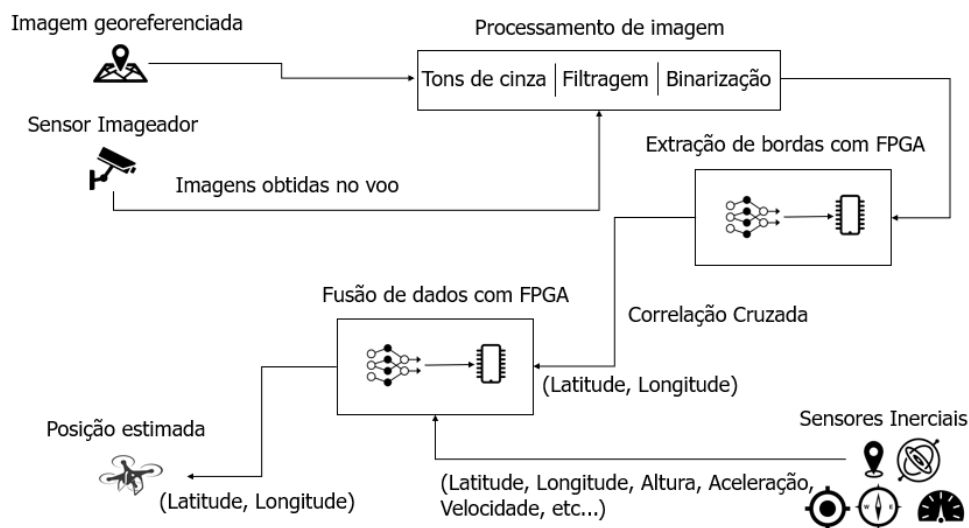
## 4 METODOLOGIA

Nesta Tese, é proposta uma abordagem para fusão de dados, aplicada à navegação autônoma de VANT, realizada por uma MLP autoconfigurada e construída em FPGA. A metodologia foi baseada no trabalho desenvolvido por CONTE e DOHERTY (2009).

A diferença estabelecida, em relação ao trabalho de CONTE e DOHERTY (2009), é que foram utilizadas a odometria para estimação da posição do VANT por imagens, FK como fusor de dados e todos os dados foram aplicados à fusão de dados em CPU - não foram realizados testes em hardware dedicado, tampouco a construção da metodologia em algum dispositivo FPGA.

Para aplicação da metodologia proposta utilizou-se dados reais, coletados de voos feitos em duas regiões, uma na Suécia e outra no Brasil. Coletou-se dados de sensores inerciais e dados (imagens) de um sensor imageador óptico. Estes dados foram submetidos a fusão de dados por uma MLP autoconfigurada e como resultado obteve-se uma estimação melhorada da posição do VANT utilizado, especificamente, o que se obteve foram medidas relativas as coordenadas latitude e longitude. Os dados e VANTs usados são apresentado no Capítulo 5. A Figura 4.1, ilustra a metodologia proposta.

Figura 4.1 - Metodologia, numa visão geral.

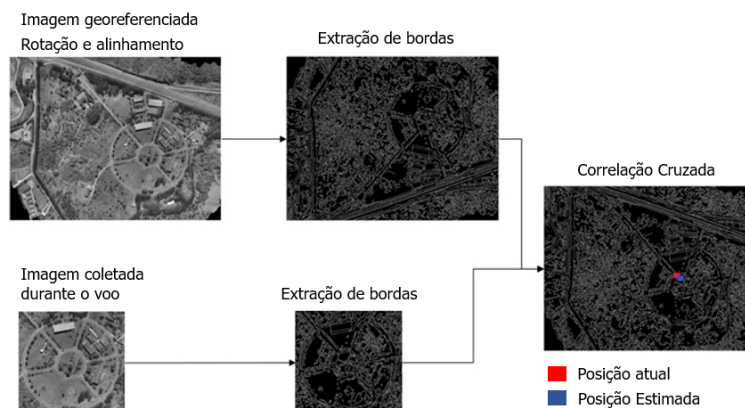


Fonte: Produção do autor.

Na Figura 4.1, é mostrada uma visão geral da metodologia na forma de fluxo. A primeira parte da metodologia consiste em coletar imagens vindas do sensor imageador, durante o voo. Estas imagens são processadas e suas bordas extraídas. Após isto, cada imagem é correlacionada com uma outra, previamente embarcada e georeferenciada. O resultado da correlação é uma primeira estimativa da posição do VANT e esta estimativa é aplicada como entrada no fusor neural. Nesta metodologia, utilizou-se uma MLP auto-configurada, também implementada em FPGA, para extração de bordas - esta abordagem foi empregada por BRAGA (2018), como explicado no Capítulo 3. A rede sem pesos de conexão aplicada para extração de bordas (VITOR TORRES et al., 2020) foi também implementada em FPGA.

Antes do processo de correlação também é necessário fazer algumas transformações nas imagens, como rotação e alinhamento. Isto é necessário para que seja possível a etapa de correlação, dado que, no momento da coleta das imagens, o VANT pode estar com sua frente apontada para qualquer direção. A Figura 4.2 ilustra o processo, de transformação e correlação, resumidamente.

Figura 4.2 - Transformação das imagens e primeira estimativa.



Fonte: Produção do autor.

A primeira estimativa da posição por correlação de imagens juntamente com os dados dos sensores inerciais formam o vetor de entradas para o fusor neural. O fusor neural é responsável por correlacionar estes dados, mitigando erros e aplicando importância aos dados mais relevantes. Tudo isto é feito num processo de mapeamento não-linear. O resultado é que o fusor neural entrega uma nova estimativa da posição do VANT.

No Capítulo 3, foram mencionadas as classificações para interação entre sensores,

na fusão de dados. Portanto, cabe dizer que na metodologia proposta o que ocorre é uma combinação das classificações, uma vez que foram utilizados dados redundantes e complementares. Por exemplo, considera-se que as informações de latitude e longitude, obtidas pela correlação de imagens e pelo INS, são dados redundantes. Já dados de aceleração e velocidade são dados complementares.

Além disso, a fusão de dados aplicada na metodologia é feita em dois níveis, como também mencionado no Capítulo 3. Isto porque ocorre um processamento sobre os dados do sensor imageador. Em outras palavras, estes dados não são oferecidos em sua forma bruta, como entrada para o fusor neural.

A métrica escolhida para comparar o desempenho da metodologia foi o erro médio da estimação de cada coordenada e da trajetória. Não há, na literatura, uma definição de qual métrica seja a melhor para comparar o desempenho de um fusor de dados aplicado à navegação autônoma de VANT. Há trabalhos que usam o erro médio quadrático (PAULINO, 2019), outros que usam um erro médio baseado nas características dos sensores embarcados (CONTE; DOHERTY, 2009). Escolheu-se o erro médio, porque junto dele também foram calculadas medidas que denotam intervalo de confiança, como desvio padrão e variância. Também, para comparação é definida uma fonte como referência, sendo que para os testes com os dados da Suécia a referência foi um FK e para os testes no Brasil a referência foi GPS-RTK.

O fusor neural consiste de uma MLP, que precisa ser treinada. Com relação ao treinamento, torna-se importante dizer que a escolha dos dados para treinamento é algo que afeta o desempenho de uma metodologia de fusão de dados baseada em aprendizagem de máquina (KHALEGHI et al., 2013). Isto acontece porque os dados de voo possuem características diferentes, o que os torna complexos à medida que a escolha dos conjuntos de dados para treinamento e teste não é algo trivial (PAULINO, 2019). Desta forma, nesta Tese, utilizou-se abordagens de particionamento dos dados baseadas em validação cruzada.

Na literatura, considera-se a validação cruzada como uma técnica para medir a capacidade de generalização de um modelo de aprendizagem de máquina, a partir de um conjunto de dados (YADAV; SHUKLA, 2016; RABINOWICZ; ROSSET, 2020). Entende-se generalização como a capacidade do modelo de responder corretamente a dados apresentados que não foram utilizados durante o treinamento. Por isso esta técnica é amplamente aplicada a problemas onde o objetivo é predição ou classificação (RABINOWICZ; ROSSET, 2020).

A forma como a validação cruzada ocorre está relacionada com a abordagem aplicada no particionamento dos dados de treinamento e validação (YADAV; SHUKLA, 2016). Uma das abordagens aplicadas ao particionamento de dados é a *hold-out* (WITTEN et al., 2011; NURHAYATI et al., 2014; YADAV; SHUKLA, 2016). Em sua forma canônica, consiste em separar os dados em duas partes, dois conjuntos, não necessariamente iguais em tamanho e mutuamente exclusivos. O treinamento do modelo de inteligência artificial acontece com um dos conjuntos e após os ajustes dos parâmetros de treinamento, que, no caso de uma MLP, são os pesos sinápticos. Em seguida, a rede neural é avaliada utilizando o conjunto restante - o conjunto para teste ou validação.

Para problemas onde se utiliza redes neurais do tipo MLP, a aplicação da abordagem *hold-out* consiste em avaliar a validação de várias redes candidatas (WITTEN et al., 2011; NURHAYATI et al., 2014; KISELEV, 2015; YADAV; SHUKLA, 2016; RABINOWICZ; ROSSET, 2020). A rede que apresentar o melhor resultado sobre o conjunto de validação é selecionada como a melhor - isto acontece durante a fase de treinamento.

Contudo, considerando a validação cruzada, é possível que a rede selecionada se sobreajuste ao conjunto de validação. Por isso, uma forma de melhorar a estimativa da abordagem é considerar a divisão dos dados em três partes (YADAV; SHUKLA, 2016). A primeira parte consiste dos dados para treinamento e a segunda parte dos dados usados para validação, semelhante a abordagem canônica. A terceira parte é denominada de conjunto de generalização e é sobre esta parte que a rede, enfim, é avaliada sobre sua capacidade de acerto. A implementação do MPCA utilizado nesta pesquisa aplica a abordagem *hold-out* de separação dos dados - considerando três conjuntos. Entretanto, também foi testada outra forma de validação cruzada, denominada de *k-fold* (YADAV; SHUKLA, 2016; RABINOWICZ; ROSSET, 2020).

A abordagem *k-fold* consiste de uma técnica de validação cruzada intensiva (WITTEN et al., 2011; NURHAYATI et al., 2014; YADAV; SHUKLA, 2016). Nesta abordagem, são usadas todas as amostras disponíveis para treinamento e validação, por isso os resultados obtidos com este tipo de validação cruzada tendem a ser melhores, principalmente quando comparados a abordagem *hold-out* (YADAV; SHUKLA, 2016). Outro ponto importante que pode ser destacado, é que com esta abordagem é possível identificar quais os conjuntos de treinamento e teste que proporcionam o melhor e o pior resultado de generalização.

Em sua forma canônica a abordagem *k-fold*, consiste em dividir o conjunto total de dados em  $k$  subconjuntos - o valor de  $k$  é definido arbitrariamente, geralmente depende do conhecimento de um especialista sobre os dados (RABINOWICZ; ROSSET,

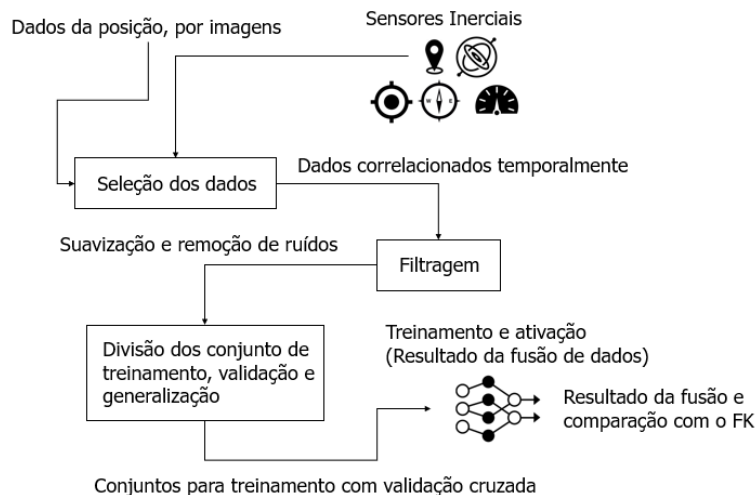
2020). Após a divisão em  $k$  subconjuntos, um subconjunto é separado para validação do modelo de aprendizagem de máquina e os conjuntos restantes são utilizados como treinamento. O processo de validação cruzada é então repetido  $k$  vezes, de modo que cada um dos  $k$  subconjuntos sejam utilizados uma vez como teste para generalização do modelo. O resultado final da validação  $k$ -fold é o desempenho médio obtido nos  $k$  testes (YADAV; SHUKLA, 2016; RABINOWICZ; ROSSET, 2020).

#### 4.1 Fases de experimentos

Nesta Tese, os experimentos foram divididos em fases. Cada fase foi definida sobre um dos conjuntos de dados, ou seja, uma fase abordou experimentos sobre os dados da Suécia e outra fase foi desenvolvida para os experimentos com os dados do Brasil.

A fase que utiliza os dados da Suécia tem por objetivo verificar se a MLP autoconfigurada é capaz de emular o FK, aplicado à fusão de dados e navegação autônoma de VANT. Durante esta fase, foram avaliados os estudos de treinamento com validação cruzada e também os estudos sobre a presença de ruídos nos dados inerciais. A Figura 4.3, mostra o fluxo de atividades utilizado para realizar os experimentos da primeira fase, com os dados da Suécia.

Figura 4.3 - Fluxo de atividades para os experimentos da primeira fase.



Fonte: Produção do autor.

Um ponto importante, é que na fase de experimentos com os dados da Suécia não se trabalhou com processamento de imagens. Isto porque os resultados do proces-

samento foram fornecidos por trabalhos anteriores (BRAGA et al., 2016; BRAGA et al., 2018; BRAGA et al., 2019). Então, foram entregues os resultados da estimação da posição por imagem, com dados de latitude e longitude. Por isso, a concentração dos experimentos sobre os dados inerciais.

A segunda fase de experimentos foi realizada no Brasil. Em particular, é interessante testar a metodologia no Brasil, pois é onde há uma alta probabilidade de falha do sinal GPS, devido a fenômenos naturais, como mencionadono Capítulo 3.

Na segunda fase, foram coletadas imagens durante o voo do VANT e estas imagens passaram pelas etapas de processamento, descritas anteriormente. Nesta fase, está a aplicação completa da metodologia, ocorrendo a construção do fusor neural com FPGA. Em acréscimo, também é construído o extrador de bordas com FPGA. Então, o fluxo de atividades desta etapa já foi descrito na Figura 4.1.

Com relação a construção das MLPs com FPGA, no Capítulo 7 é descrito cada passo da construção do *hardware* dedicado. Foram testadas três implementações do fusor neural com FPGA, em dois dispositivos diferentes, e a diferença entre elas foi a forma de construção da função de ativação da MLP. A cada implementação foram evidenciados os resultados da fusão de dados e suas respectivas incertezas, sendo possível indicar o impacto quando a fusão de dados é realizada com FPGA.

Além da implementação da metodologia completa, com fusão de dados em FPGA, foi incorporada à metodologia a estimação das incertezas durante o voo. Os trabalhos usados como referência também mostram incertezas associadas as suas estimações. Porém, estas incertezas são calculadas em CPU, em solo, apenas para avaliação da fusão de dados. Destaca-se que não é ofertado um método para estimar incertezas durante o voo. Isto indicaria ao computador de bordo ou a um *software* de controle embarcado se a fusão está correta ou não. Portanto, na literatura, ainda não é abordado um sistema ou técnica de informação, caso ocorra falha na fusão de dados, mas, nesta Tese, isto é considerado. Como isto ocorreu sobre os mesmos dados da segunda fase de experimentos, esta etapa não foi considerada como uma terceira fase.

Por fim, além da estimação das incertezas, também foi realizada à estimação da posição do VANT por Filtro de Partículas. Desta forma, foi possível comparar a diferença entre a estimação, com fusor neural, em FPGA e a estimação por uma técnica bayesiana. Tanto a estimação da posição por Filtro de Partículas, quanto a estimação das incertezas são demonstradas no Capítulo 8.

## 5 DADOS

Para validação da metodologia proposta foram usados dados de voos reais. Utilizou-se dois tipos de VANTs, com voos em duas regiões diferentes.

### 5.1 Dados do WITAS

O primeiro conjunto de dados utilizado foi obtido dos experimentos realizados pelo grupo de pesquisa do WITAS (*Wallenberg Laboratory for Information Technology and Autonomous Systems*) na universidade de Linköping - Suécia (DATTA-BARUA et al., 2003; CONTE; DOHERTY, 2008; CONTE; DOHERTY, 2009). A abordagem proposta pelo WITAS serviu de base para os primeiros experimentos realizados nesta pesquisa.

Nos experimentos feitos pelo WITAS, utilizou-se um VANT autônomo, baseado no modelo comercial Yamaha R-MAX (YAMAHA MOTORSPORTS, 2020). Este VANT é mais robusto do que o utilizado nos experimentos no Brasil, dado que sua capacidade de carregar peso é maior. O comprimento total do VANT é de cerca de 3,6m e com capacidade de decolar com o peso máximo de 95kg (CONTE; DOHERTY, 2009; YAMAHA MOTORSPORTS, 2020). A Figura 5.1 mostra o modelo utilizado nos experimentos na Suécia.

Figura 5.1 - VANT utilizado nos experimentos na Suécia.



Fonte: Adaptado de CONTE e DOHERTY (2009).

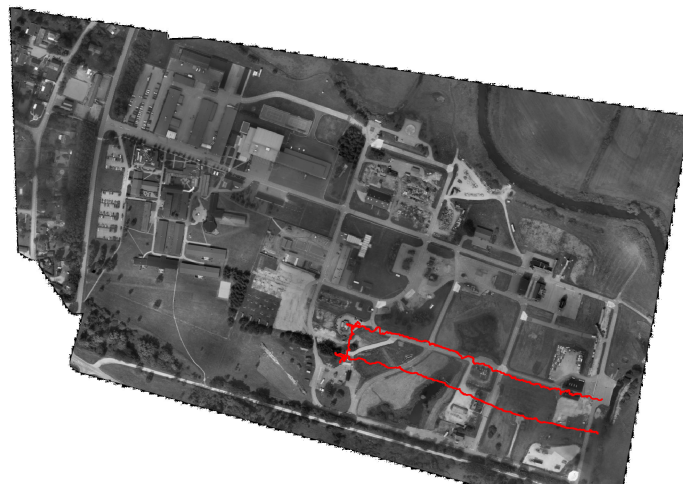
A tecnologia aplicada na construção do VANT foi desenvolvida pelo Departamento



de Informática e Ciência da Informação na Universidade de Linköping (CONTE; DOHERTY, 2009). Os sensores utilizados foram embarcados em uma IMU, contendo três acelerômetros e três giroscópios que forneceram a aceleração e a taxa angular do VANT ao longo dos três eixos. Foram utilizados um sensor barométrico de altitude e uma câmera de vídeo monocular montada em uma unidade de panorâmica. Junto com estes sensores, também foram obtidos as medidas de latitude e longitude vindas do INS.

O sistema de controle do VANT foi realizado por três computadores embarcados (CONTE; DOHERTY, 2009). O computador de voo, foi um PC104 PentiumIII 700MHz. Este computador foi responsável por enviar os comandos para decolagem, aterrissagem e demais movimentos básicos durante o voo. O segundo computador, também do mesmo modelo que o primeiro, implementou o processamento de imagens e funcionalidades que controlavam a inclinação da câmera. Por fim, o último computador, um PC104 Pentium-M 1,4GHz, implementou a fusão de dados - neste computador ocorreu o processamento e resultado do FK. Toda a comunicação interna entre os computadores e a malha de controle do VANT foi realizada por conexões do tipo IEEE 802.3 (Ethernet) (CONTE; DOHERTY, 2009). A Figura 5.2 mostra a trajetória realizada pelo VANT, utilizado nos experimentos na Suécia.

Figura 5.2 - Trajetória realizada pelo VANT nos experimentos na Suécia.



A trajetória percorrida é denotada pela linha em vermelho.

Fonte: Adaptado de CONTE e DOHERTY (2009).

Durante a trajetória foram coletados 12 tipos de dados que são latitude e longitude



obtidas pelo sensor inercial, que consistiram em 3 taxas angulares, 3 medidas de acelerômetro, medida de velocidade norte, oeste, subida e altura. Estas informações, juntamente com a estimação por processamento de imagens, foram aplicadas ao FK para estimação da posição do VANT (CONTE; DOHERTY, 2009). Estes dados foram armazenados nos *logs* de voo em uma estrutura tabular com 66.483 amostras temporais.

Os dados foram armazenados, considerando a taxa de amostragem mais rápida dos sensores giroscópio e acelerômetro. Isto gerou, em vários momentos, informações repetidas. Por isso, nas etapas de treinamento do Capítulo 6, é aplicado um tratamento que provocou uma redução para 3.319 amostras temporais, sem criar dados novos. Este conjunto reduzido foi aplicado nas etapas de treinamento e validação cruzada e também nas avaliações, considerando filtragem.

## 5.2 Dados do IEAv

O segundo conjunto de dados utilizado foi obtido em um voo feito no Brasil. O voo foi realizado no Instituto de Estudos Avançados (IEAv), que fica em São José dos Campos - SP.

Nos experimentos feitos no IEAv, utilizou-se um VANT do modelo comercial DJI Matrice 600 (DJI, 2020a). Para este VANT, a capacidade de peso máximo para decolagem é de 15kg. Apesar de ser menos robusto do que o VANT usado nos experimentos da Suécia, a fuselagem desta aeronave está equipada com tecnologias mais recentes e melhores - sensores com melhores precisões (PAULINO, 2019; DJI, 2020a). A Figura 5.3 mostra o VANT utilizado nos experimentos no Brasil.

Diferentemente do VANT utilizado nos experimentos da Suécia, que foi construído pela equipe técnica do WITAS, o DJI Matrice 600 aplicado nos experimentos no Brasil não foi construído por uma equipe técnica do IEAv. Na realidade, este VANT foi utilizado com as configurações e componentes originais fornecidas pelo fabricante.

O DJI Matrice 600 possui uma controladora de voo embarcada, que reúne e processa informações dos sensores acoplados ao corpo da aeronave (DJI, 2020a). A controladora é responsável por gerenciar a malha de controle do VANT e também por trocar informações com a estação de solo, quando a navegação está sob gerência de um piloto humano (DJI, 2020a). A forma de comunicação entre o a estação de solo e o VANT é por uma rede sem-fio do tipo IEEE 802.11 (DJI, 2020a).

Algumas características do DJI Matrice 600 são: velocidade horizontal máxima de

Figura 5.3 - VANT utilizado no experimento no Brasil.



Fonte: Adaptado de DJI (2020a).

65km/h, velocidade máxima de subida de 5m/s e velocidade máxima de descida de 3m/s (DJI, 2020a). Este VANT também pode chegar a altitude máxima de 2.500m (DJI, 2020a).

Esta aeronave também possui um equipamento que é formado de três sensores do tipo GPS-RTK. Estes sensores atuam de maneira redundante, entregando a controladora medidas de latitude e longitude. A média dessas informações garante uma precisão de 2cm no posicionamento do DJI Matrice 600, em relação a um marco em solo georeferenciado (PAULINO, 2019; DJI, 2020a). Esta precisão é significativamente melhor do que a que foi obtida pelos sensores do VANT usado na Suécia (CONTE; DOHERTY, 2009; PAULINO, 2019).

Além do conjunto de sensores GPS-RTK, o modelo DJI Matrice 600 pode capturar, durante sua navegação, 315 tipos diferentes de dados (PAULINO, 2019; DJI, 2020a). Dentre esses tipos, estão as mesmas informações obtidas pelo VANT do WITAS, como: medidas de latitude e longitude fornecidas pelo sensor INS, altura, taxas angulares e taxas de aceleração. Junto destas informações também estão disponíveis para uso sensores como magnetômetro e barômetro. A eletrônica do DJI Matrice 600 também entrega a controladora informações sobre a velocidade de rotação dos motores e outras informações dos componentes embarcados.

Com esta aeronave, foi realizado um voo sob uma área de aproximadamente 200m<sup>2</sup>. A trajetória do voo realizado seguiu o formato de um hexágono. Esta trajetória pode ser vista na Figura 5.4. Cabe ressaltar as seguintes informações sobre o voo realizado:





## 6 PRIMEIRA FASE DE EXPERIMENTOS - DADOS DO WITAS

Neste Capítulo são apresentados os primeiros experimentos e resultados realizados para verificar a validação da metodologia proposta, sobre os dados do grupo de pesquisa do WITAS. Foi utilizado um conjunto de dados, com informações reais de voo de VANT - obtidos pela aeronave utilizada na Suécia. Optou-se por utilizar esses dados porque já foram estudados em outros trabalhos com outros tipos de técnicas para fusão de dados (CONTE; DOHERTY, 2008; CONTE; DOHERTY, 2009; BRAGA et al., 2016; PAULINO, 2019).

O principal objetivo deste Capítulo é investigar os resultados de uma MLP autoconfigurada, aplicada a fusão de dados, como técnica substituta ao FK. Para aferir a efetividade e confiabilidade da metodologia proposta, foram realizados estudos que foram divididos nas seguintes fases:

- Treinamento e validação;
- Avaliação ao aplicar filtragem;
- Combinação dos sensores de atitude.

As incertezas mostradas neste Capítulo foram calculadas sobre a diferença entre o que foi obtido pela MLP autoconfigurada e por uma fonte de referência. Neste Capítulo, a fonte de referência é o FK, mais precisamente, o resultado da fusão de dados quando aplicado o FK como fusor. Este FK foi desenvolvido pelo grupo de pesquisa do WITAS (CONTE; DOHERTY, 2008; CONTE; DOHERTY, 2009).

Uma vez que foram utilizados os dados do grupo de pesquisas do WITAS, torna-se importante informar que o erro médio na trajetória entre as medidas de latitude e longitude, obtidas pelo INS e a referência usada nos experimentos na Suécia, é de aproximadamente 10m (CONTE; DOHERTY, 2008; CONTE; DOHERTY, 2009; BRAGA et al., 2016; BRAGA, 2018). A fusão de dados, feita pela FK, reduziu essa diferença para menos que 5m (CONTE; DOHERTY, 2008; CONTE; DOHERTY, 2009; BRAGA et al., 2016; BRAGA, 2018).

### 6.1 Treinamento e validação

É comum que os sensores embarcados em um VANT tenham taxas de aquisição diferentes (FRADEN, 2010; KHALEGHI et al., 2013; VALAVANIS; VACHTSEVANOS, 2014). Isto acontece porque, geralmente, a fabricação de cada sensor não é feita por um só

fornecedor de componentes. Acrescenta-se que os canais de comunicação usados na eletrônica embarcada podem variar em tipo e tecnologia, cada um com uma taxa de transferência própria. A variação nas taxas de aquisição também ocorreu com os dados do WITAS.

Dado que os sensores possuem taxas de aquisição diferentes, os tempos de coleta de dados foram desunidos - um dos metadados, no log de voo, é o tempo que a informação foi registrada pelo sensor, que é medido em milissegundos logo após os motores da aeronave serem ligados. O FK embarcado no VANT usado na Suécia possuía uma taxa de ativação de 50Hz. Já os sensores giroscópio e acelerômetro possuíam uma taxa de aquisição de 200Hz. Isto fez com que as medidas obtidas pelo sistema fossem repetidas em vários momentos. Portanto foi necessário uma pré-seleção nos dados, para interpolar as medidas obtidas pelos sensores de maneira que elas fossem relacionadas temporalmente.

Os dados originais estavam organizados em 66.483 amostras temporais. Então, tomando como base o sensor com menor taxa de aquisição, com 10Hz, uma pré-seleção foi realizada sobre os dados. Esta pré-seleção significou desprezar as medidas de tempo descasadas, considerando o sensor com menor taxa de aquisição. Deste modo, os dados foram reorganizados em um conjunto de 3.319 amostras temporais. Estes dados foram usados para o treinamento e validação da MLP autoconfigurada. Nos experimentos foram utilizados 12 tipos de dados que são latitude e longitude obtidas pelo sensor inercial, sendo 3 taxas angulares, 3 medidas de acelerômetro, medidas de velocidade norte e oeste, subida e, por último, altura.

Um dos objetivos do trabalho é a implementação do método de fusão de dados (KHALEGHI et al., 2013), em *hardware* dedicado com FPGA. Para implementação de uma MLP em FPGA, é necessário considerar que o número de neurônios e camadas pode aumentar a complexidade do desenvolvimento de maneira significativa. Além disto, as placas usadas para implementação possuem uma limitação de capacidade e, dependendo do tipo de *chip* fornecido pelo fabricante, poucos neurônios já são suficientes para ocupar o espaço eletrônico configurável.

Neste trabalho, utilizou-se uma implementação do MPCA (ANOCHI et al., 2015), que consiste em um *software* que pode ser parametrizado. Entre os parâmetros a serem definidos estão a variação na taxa de aprendizagem da rede neural, a variação no número de épocas de treinamento, a variação na taxa de momento, os tipos da função de ativação, a variação na quantidade de neurônios por camada e a variação no número de camadas escondidas. Para a quantidade de neurônios a variação se-

lecionada foi de 1 até 32 neurônios - esta quantidade foi selecionada após avaliar a capacidade da eletrônica configurável na placa *LOGI Pi* (VALENTFX, 2020), usada para implementação da MLP em FPGA.

### 6.1.1 Treinamento com validação cruzada - *hold-out*

Alguns estudos abordam a relação entre os dados disponibilizados pelo grupo de pesquisa do WITAS e a sua divisão em grupos para treinamento de um modelo de aprendizagem de máquina. Na literatura, encontra-se um trabalho que investigou a influência do número de épocas de treinamento sobre este conjunto de dados. Ressalta-se que a metodologia aplicada na investigação não foi uma MLP, mas uma abordagem utilizando *Fuzzy* (PAULINO, 2019). A investigação conclui que um baixo número de épocas de treinamento, como menor que 50 épocas, não possui segurança estatística para garantir qualidade na estimação. Além disto, verificou-se também a influência quando se varia a quantidade de dados usados no conjunto de treinamento. Os melhores resultados foram no intervalo entre 60% e 80% dos dados.

A partir disto, considera-se que já houve uma redução no total de dados para alinhamento temporal e que usar 80% do total de dados pode não ser interessante por causa de *overfitting*. Então, neste experimento, os dados foram separados da seguinte forma, os dois primeiros conjuntos consistiam de 70% dos dados para treinamento e 20% para validação. Os demais 10% foram usados para testes de generalização. Os erros para o desempenho da topologia encontrada pelo MPCA foram calculados sobre o conjunto de 10% dos dados.

Neste experimento o intervalo na variação para as taxas de aprendizagem e momento foi de 0,1 até 0,9. O número de épocas foi variado de 100 até 300. Dado que os valores de latitude e longitude não são valores discretos, testou-se as funções de ativação tangente hiperbólica e sigmóide - não se considerou funções de ativação, como a degrau, porque existe uma limitação nos tipos de funções de ativação que se pode usar com a implementação do MPCA aplicado no experimento. As informações na Tabela 6.1 são os parâmetros de treinamento, encontrados pelo MPCA, que apresentaram o melhor resultado sobre o conjunto de generalização.

Os erros encontrados para as coordenadas latitude e longitude, neste experimento, foram respectivamente de 47,57cm e 62,83cm. A Tabela 6.2 apresenta as incertezas encontradas sobre o conjunto de generalização, para cada coordenada. O erro médio na trajetória foi estimado em 87,38cm - considera-se aqui o deslocamento real, o valor obtido a partir das estimações feitas pelo FK. Sobre o erro da trajetória, considera-se

Tabela 6.1 - Parâmetros selecionados pelo MPCA para o primeiro experimento.

Parâmetro	Valor ou Tipo
Épocas de treinamento	300
Neurônios na camada escondida	11
Taxa de Aprendizagem	0,7
Taxa de momento	0,8
Função de ativação	sigmóide

que cada localização no espaço é definida por um par do tipo latitude e longitude, como em um plano cartesiano. Desta forma, existe uma distância associada à dois pontos, que, no caso deste experimento, corresponde à diferença no deslocamento real e no deslocamento estimado pela MLP. Para uma trajetória existe um conjunto de distâncias entre os pontos reais e os pontos estimados, sendo considerado o erro na trajetória a média destas distâncias.

Tabela 6.2 - Incertezas em centímetros para cada coordenada - primeiro experimento.

Latitude	
Tipo de incerteza	Valor
Erro médio	47,57cm
Desvio	45,74cm
Variância	2.092,44cm
Longitude	
Tipo de incerteza	Valor
Erro médio	62,83cm
Desvio	49,46cm
Variância	2.447,09cm

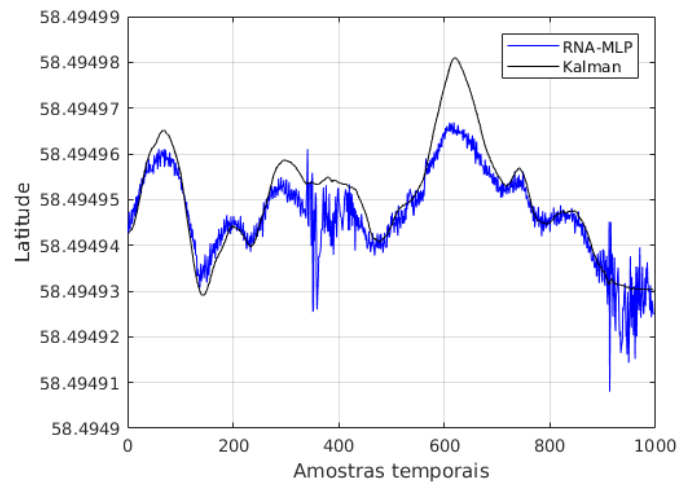
Neste experimento, os valores obtidos de desvio e variância foram altos, o que pode ser indicativo de sensibilidade ao ruído. Para medidas de latitude e longitude, uma variação em graus, ainda que pequena, corresponde à uma variação em metros. Por exemplo, a distância de 1° em latitude equivale a aproximadamente 111,12km (YING-KUN, 2018). Desta forma, qualquer variação nas medidas de latitude e longitude, ocorridas durante o voo, quando convertidas para metros corresponderam a valores significativos.

Na Figura 6.1 mostra-se o resultado da estimação aplicada aos conjuntos de validação e generalização. É importante ressaltar que os valores de latitude e longitude mostrados na Figura 6.1 estão em grau e que a variação nestas medidas ocorre na

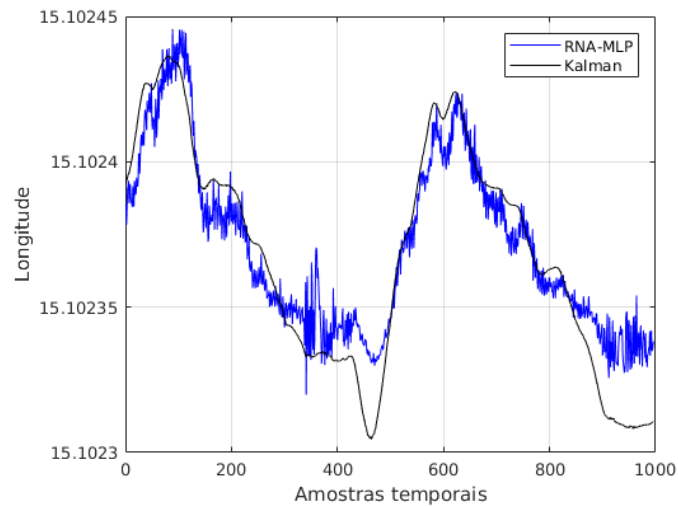


parte da mantissa para toda a navegação do VANT. Isto é esperado para este tipo de dado, em que geralmente a variação ocorre a partir da quinta casa decimal, considerando que a área de voo não foi de grande tamanho. Por isso, pode-se causar estranheza ao olhar os valores da Tabela 6.2, que estão em centímetros, e os da Figura 6.1, que estão em grau.

Figura 6.1 - Resultado da estimação sobre os conjuntos de validação e generalização, para o primeiro experimento.



(a)



(b)

(a) Valores para a coordenada de latitude. (b) Valores para a coordenada de longitude.

Fonte: Produção do autor.

Mesmo com um trabalho apontando que é melhor para o treinamento usar entre 60% e 80% do total de dados, decidiu-se usar o benefício da dúvida. Desta forma, um outro experimento foi realizado com a intenção de investigar a influência do percentual de dados para o treinamento. A diferença foi uma redução do conjunto de treinamento que passou a ser de 30%. O objetivo foi investigar se com um percentual menor obtém-se resultados próximos ao experimento anterior. Os demais conjuntos foram divididos em 60% para validação e manteve-se o mesmo conjunto de 10% para generalização. Ainda manteve-se também a mesma variação na quantidade de épocas de treinamento, os demais parâmetros de treinamento foram variados, como no experimento anterior.

A escolha em usar 30% dos dados foi determinada a partir do intervalo considerado como o melhor para treinamento. Dado que o total de dados disponíveis são de 3.319 amostras temporais, se fossem usados apenas 10% dos dados para treinamento seria pouco. Já um subconjunto de treinamento com 50% dos dados ficaria próximo ao intervalo considerado como melhor - que inicia em 60%. Por isso decidiu-se por uma porcentagem intermediária.

A Tabela 6.3 mostra os parâmetros de treinamento encontrados pelo MPCA. Houve alteração nos parâmetros de treinamento encontrados, mas uma coisa a se destacar é a quantidade de neurônios na camada escondida que foi para 23, um aumento de quase 50%. Sobre o aumento do número de neurônios, cabe ressaltar que o MPCA possui natureza estatística (ANOCHI et al., 2015). Então, a cada execução é comum obter uma nova configuração de rede, isto significa que a convergência não necessariamente será sobre uma mesma arquitetura.

Neste outro experimento, a diferença das estimações para cada coordenada também foi significativa. Os erros encontrados para as coordenadas latitude e longitude foram, respectivamente, de 74,70cm e 111,41cm - destaca-se a longitude, onde o erro médio passou de 1m. A Tabela 6.4 mostra as incertezas calculadas. O aumento do erro para cada coordenada foi maior que 50% em relação aos valores apresentados na Tabela 6.2. Conclui-se que usar uma quantidade menor para o conjunto de treinamento gerou resultados piores.

### **6.1.2 Treinamento com validação cruzada - *k-fold***

Nos experimentos anteriores os conjuntos de dados foram separados de forma sequencial no tempo. Por isso, os últimos 10

Tabela 6.3 - Parâmetros selecionados pelo MPCA para o segundo experimento.

Parâmetro	Valor ou Tipo
Épocas de treinamento	300
Neurônios na camada escondida	23
Taxa de Aprendizagem	0,6
Taxa de momento	0,7
Função de ativação	tangente hiperbólica

Tabela 6.4 - Incertezas em centímetros para cada coordenada - segundo experimento.

Latitude		
Tipo de incerteza	Valor	Aumento na incerteza
Erro médio	74,70cm	57,03%
Desvio	56,09cm	22,62%
Variância	3.146,57cm	50,37%
Longitude		
Tipo de incerteza	Valor	Aumento na incerteza
Erro médio	111,41cm	77,31%
Desvio	62,83cm	27,03%
Variância	3.947,87cm	61,32%

Contudo, esta forma de divisão dos dados pode ser ineficiente para o treinamento, uma vez que não se sabe em qual parte dos dados está o conteúdo mais significativo, ou seja, em qual parte estão as informações que seriam as melhores para treinar a rede neural. Além disso, treinar uma rede neural utilizando dados sequenciais pode gerar uma aprendizagem viciada em tendências, o que aumenta o erro de generalização, quando o conjunto aplicado para teste não segue a mesma tendência do conjunto de treinamento. Para evitar este problema, decidiu-se, então, aplicar outra abordagem de particionamento conhecida na literatura por *k-fold* (CRISTIANINI, 2004; WITTEN et al., 2011; NURHAYATI et al., 2014; YADAV; SHUKLA, 2016; RABINOWICZ; ROSSET, 2020).

O objetivo da validação cruzada *k-fold* é investigar o desempenho a partir da variação sobre os conjuntos de treinamento e generalização. Então, considerando que para uma rede neural do tipo MLP os pesos sinápticos são inicializados de maneira aleatória, torna-se conveniente que, para todos os *k* subconjuntos, a validação ocorra partindo da mesma estrutura de rede. Logo, inicia-se uma topologia com seus respectivos neurônios, camadas escondidas e pesos sinápticos. Desta forma, para cada treinamento e validação a topologia é inicializada da mesma forma. Assim, pode-se garantir que o resultado final para cada treinamento não teve influência da iniciali-

zação aleatória dos pesos sinápticos, por exemplo.

Anteriormente, mencionou-se que, nesta pesquisa, foi aplicada uma implementação em *software* do MPCA. Um dos parâmetros que podem ser definidos para execução do *software* é a topologia inicial da rede neural. Desta forma, para cada  $k$  teste iniciou-se com a mesma estrutura de rede neural. Dado que nos experimentos anteriores o conjunto de generalização era composto de 10%, optou-se por manter esta porcentagem e, assim, foram gerados 10 novos conjuntos com aproximadamente 332 amostras temporais. A variação nos parâmetros de treinamento, como taxa de aprendizagem, taxa de momento e número de épocas também seguiram da mesma forma que nos experimentos anteriores.

O resultado obtido a partir da abordagem *k-fold* apresentou melhoras em relação aos experimentos anteriores. Apesar do erro médio ficar próximo ao obtido no primeiro experimento, houve redução sobre os valores de desvio e variância, a redução na variância sobre a latitude foi de 46%, já para longitude foi de 36%. A Tabela 6.5 mostra as incertezas obtidas a para esta abordagem de validação cruzada. O valores de redução foram calculados sobre os resultados do primeiro experimento. Para esta comparação, foi utilizada a média dos erros médios de cada conjunto.

Tabela 6.5 - Incertezas em centímetros para cada coordenada - terceiro experimento.

Latitude		
Tipo de incerteza	Valor	Redução na incerteza
Erro médio	43,92cm	7,67%
Desvio	31,28cm	31,61%
Variância	1.113,26cm	46,79%
Longitude		
Tipo de incerteza	Valor	Redução na incerteza
Erro médio	59,30cm	5,61%
Desvio	37,53cm	24,12%
Variância	1.562,86cm	36,13%

Sobre este experimento, cabe ressaltar que a divisão em 10 conjuntos ainda considerou a sequência de aquisição dos dados, ou seja, cada conjunto representa uma sequência temporal de 10% dos dados, sendo o primeiro conjunto formado pelos 332 primeiros dados e assim por diante.

Como dito anteriormente, a abordagem *k-fold* permite identificar quais conjuntos obtém o melhor e o pior resultado para generalização (YADAV; SHUKLA, 2016). Desta

forma, verificou-se o erro médio e desvio para cada um dos conjuntos. A Figura 6.2 mostra o erro médio e desvio de cada conjunto para ambas as coordenadas. Percebeu-se que alguns conjuntos obtiveram erro e desvio maior. O conjunto 6 obteve os piores resultados para erro e desvio. Isto denota que este conjunto possui uma parcela dos dados relevante e seria interessante deixar esta parcela dentro do conjunto de treinamento, o mesmo vale para os dois últimos conjuntos. Também é possível ver os conjuntos com melhor resultado para generalização. Por exemplo, o conjunto 3 obteve o menor erro médio para ambas as coordenadas e o conjunto 8 o menor de desvio, considerando as duas coordenadas.

Dado que o último conjunto, com 10% dos dados, foi utilizado para generalização nos experimentos anteriores, entende-se que o resultado do primeiro experimento foi influenciado. Então, se este conjunto estivesse na parcela de dados usada para treinamento, os resultados anteriores poderiam ser melhores.

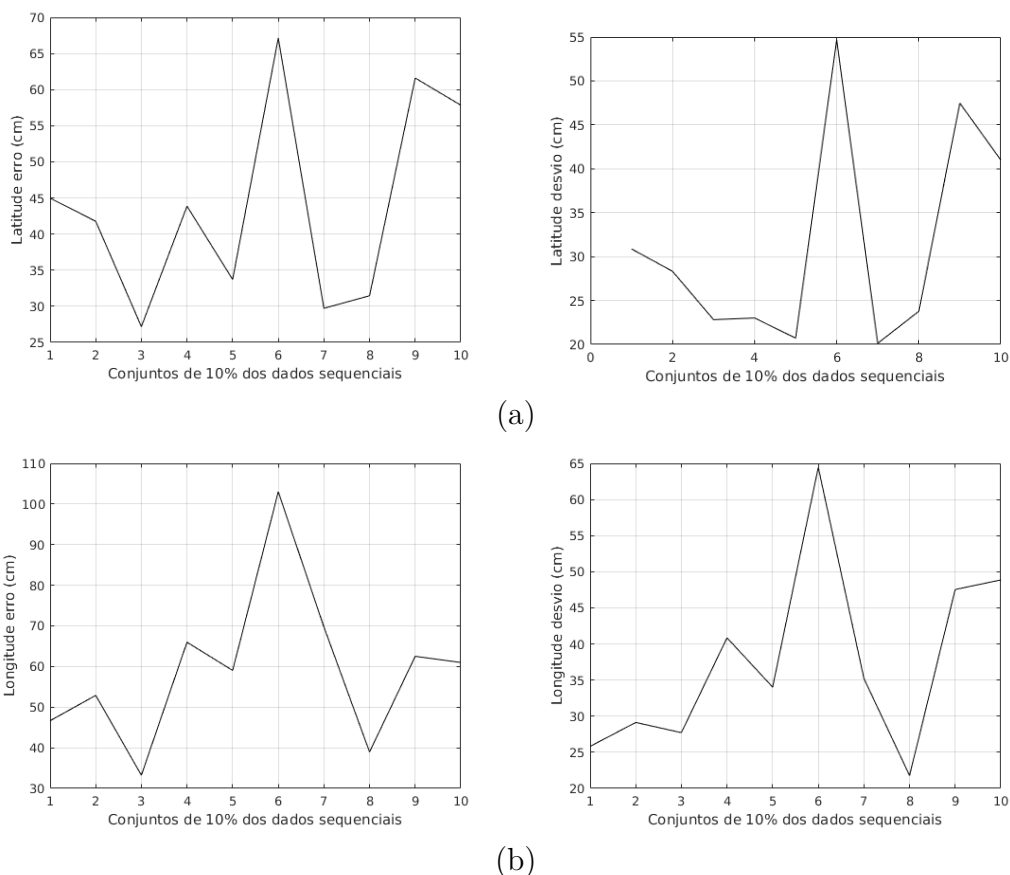
A Tabela 6.5 mostrou que houve uma redução sobre as variações - desvio e variância. Tais medidas são relacionadas, então cabe revelar que a variância foi apresentada nos experimentos para que se tenha uma informação sobre intervalo de confiança. Deste modo, o intervalo de confiança obtido em todos os experimentos foi alto, considerando o contexto da navegação autônoma de VANT. Claro que um intervalo de confiança alto não é desejado. Contudo, vale a visualização das estimativas mostradas nas Figuras 6.3 e 6.4 - os dados estão em graus. Pelas figuras percebe-se que as estimativas feitas pela rede neural seguem os valores oriundos do FK.

Em acréscimo, percebeu-se que mesmo com variações pequenas nas medidas em graus, quando se calcula os erros e variações em centímetros obtém-se valores expressivos. Isto significa que no contexto da navegação autônoma de VANT, qualquer técnica de fusão de dados deve ser treinada para responder sensivelmente a pequenas variações, para garantir um intervalo de confiança adequado.

Apesar do terceiro experimento apontar os conjuntos com relevância, ainda não foi possível determinar qual a melhor combinação entre os conjuntos para que se tenha o melhor treinamento. Além disso, a forma como os dados são organizados afeta diretamente o resultado final para generalização, principalmente quando se trata de dados complexos como os obtidos durante um voo de VANT. Isto faz com que esta tarefa não seja algo trivial. Por isso, torna-se interessante e importante definir qual é a melhor organização a partir dos dados disponíveis.

Desde o primeiro experimento, a organização dos dados seguiu utilizando 70% para

Figura 6.2 - Erro médio de generalização para cada  $k$  conjunto.

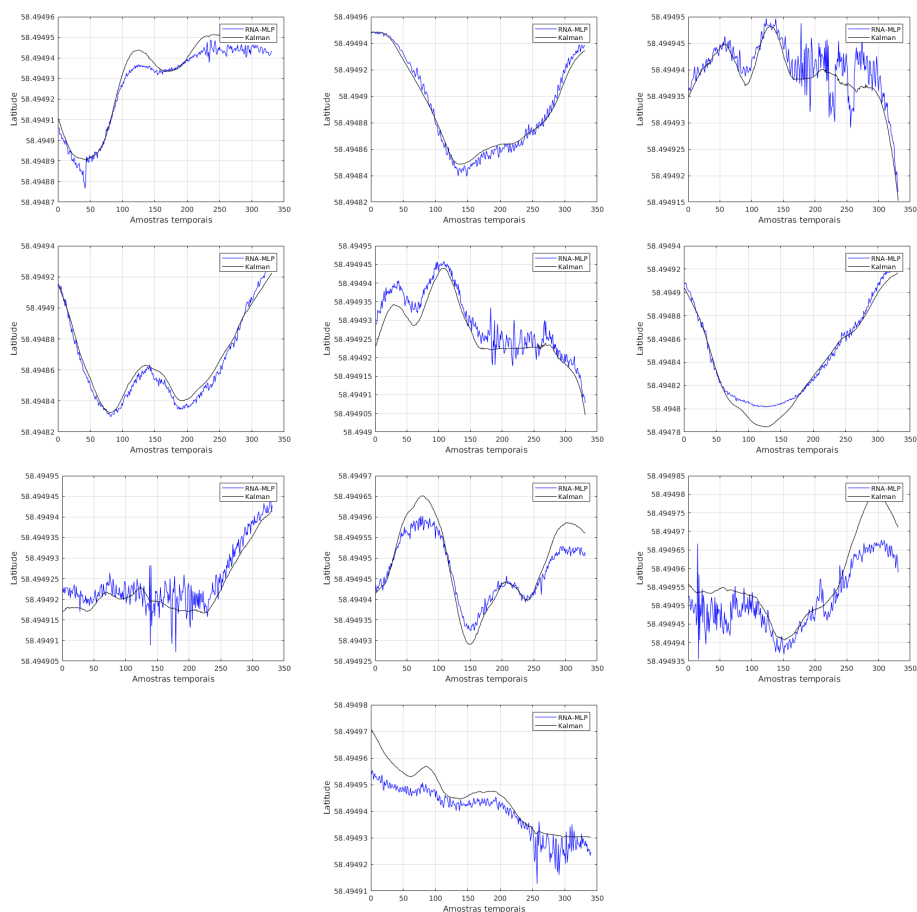


(a) mostra respectivamente o erro médio e desvio para latitude. (b) mostram respectivamente o erro médio e desvio para longitude.

Fonte: Produção do autor.

treinamento, 20% para validação e 10% para generalização. Tais porcentagens foram escolhidas porque são comuns em problemas que utilizam modelos de inteligência artificial. Entretanto, o primeiro experimento que utilizou a abordagem *k-fold*, não realizou todas as combinações possíveis entre os conjuntos de dados. Além disso, não considerou as possíveis combinações entre os conjuntos de validação e generalização - que somam 30% dos dados. Deste modo, um outro experimento foi realizado sobre os dados, que seguiu o esquema de agrupar os dados pela combinação de 10, tomados 3 a 3. Assim, manteve-se a porcentagem de 70% para treinamento, mas foram feitas variações na forma de combinação dos 30% utilizados. Portanto, nos conjuntos de validação e generalização, totalizou-se 120 combinações possíveis e independentes entre si.

Figura 6.3 - Estimações de latitude para cada  $k$  conjunto.

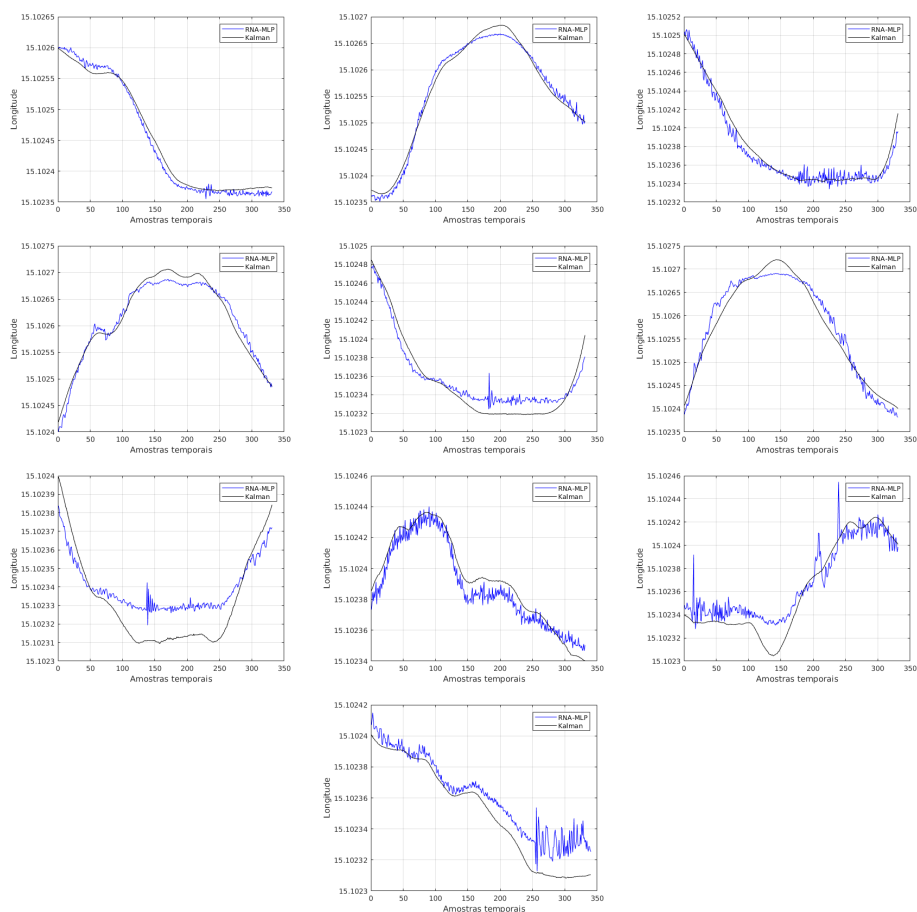


Fonte: Produção do autor.

Outro ponto a ser destacado é que no terceiro experimento foi aplicado o *software* MPCA e, como dito, este *software* iniciou o treinamento partindo da mesma arquitetura inicial. Porém, ao longo do seu funcionamento, o algoritmo troca constantemente a estrutura da rede neural, ou seja, suas camadas e neurônios sofrem influência e não mantém a mesma característica (ANOCHI et al., 2015). Desta forma não é possível garantir que o resultado obtido para cada conjunto não foi afetado por estas trocas.

Como mencionado, é interessante na abordagem *k-fold* que o treinamento sobre cada conjunto não seja influenciado com as variações na estrutura da rede neural. Por isso, neste outro experimento, não foi utilizado o *software* MPCA, mas outra plataforma que permitiu iniciar o treinamento sobre a mesma estrutura. Então, este experimento aconteceu sem variações na estrutura da MLP ao longo do treinamento, exceto o

Figura 6.4 - Estimações de longitude para cada  $k$  conjunto.



Fonte: Produção do autor.

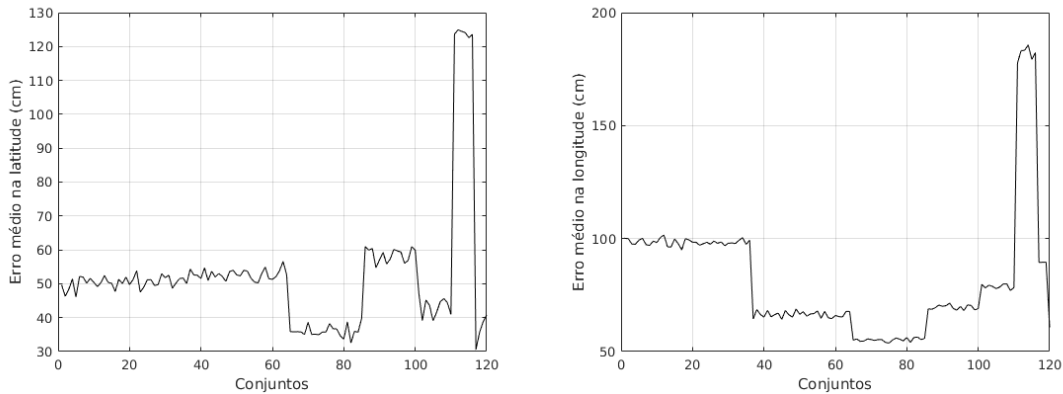
ajuste natural dos pesos sinápticos pelo método de treinamento. Assim, foi possível averiguar os conjuntos que proporcionam os menores e maiores erros possíveis para o esquema com os 120 agrupamentos formados.

Os resultados obtidos ao utilizar esta abordagem revelaram que a combinação de dados que apresenta o menor erro na generalização está no conjunto 75. Para este conjunto, aconteceu o menor erro nas estimações para a trajetória e também um dos menores erros para latitude e longitude. A Figura 6.5 mostra os erros médios nas estimações feitas para cada conjunto, para ambas as coordenadas e para trajetória. Foi possível perceber que existe um intervalo de conjuntos, iniciando acima do conjunto 60 e terminando acima do conjunto 80, que proporcionam os melhores resultados, ou seja, os menores erros médios. Ressalta-se que para latitude o melhor resultado não está neste intervalo, mas em um conjunto próximo ao último que foi

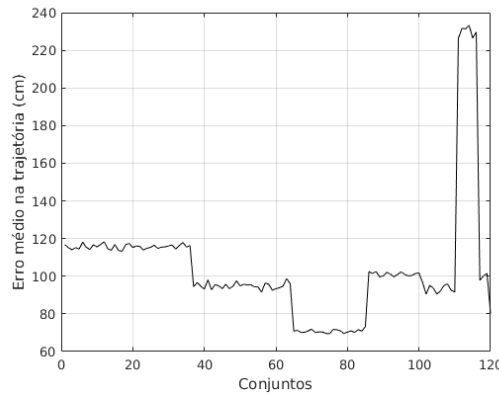


mostrado na Figura 6.5. Contudo, isto não se reflete na outra coordenada e no erro da trajetória, o que permite que este valor não seja considerado para uma possível classificação, caso seja necessário elencar os conjuntos a fim de escolher um deles como o melhor ou mais adequado.

Figura 6.5 - Estimativas de cada  $k$  conjunto, seguindo o esquema de 120 combinações.



(a)



(b)

(a) Erro para as coordenadas latitude e longitude. (b) Erro para a trajetória.

Fonte: Produção do autor.

Considerando o primeiro experimento, antes de usar a abordagem  $k$ -fold, o erro na trajetória foi de 87,28cm. Já na avaliação dos 120 conjuntos, para o conjunto 75 este erro diminuiu para 69,41cm. A redução foi 17,97cm, o que representa um erro médio 20% menor. O mesmo aconteceu com as coordenadas em que, no primeiro experimento, os erros médios para latitude e longitude foram, respectivamente, 47,57cm e 62,83cm. Enquanto que para o conjunto 75, os erros médios foram 35,74cm e 53,73cm. Isto representa uma redução de 24% e 14% para latitude e longitude.

Se for considerado a média dos valores obtidos para cada conjunto, o erro na trajetória seria de 104,04cm. Já para as coordenadas seria de 51,65cm para latitude e 81,68cm para longitude. Tais valores são consideravelmente mais altos do que os obtidos no conjunto 75 e até mesmo mais altos do que os obtidos no primeiro experimento. Isto evidencia, claramente, o impacto que pode existir com base na divisão e na forma como são selecionados os dados para compor os conjuntos de treinamento, validação e generalização.

### 6.1.3 Considerações sobre a etapa de treinamento e validação

Os experimentos realizados objetivaram o melhor entendimento sobre o conjunto de dados. Principalmente, serviram para verificar o funcionamento da metodologia sem tratamento sobre os dados, ou com o mínimo possível.

Sobre o número de épocas, a variação entre 100 e 300 foi definida a partir de observações sobre testes preliminares. Nestes testes, observou-se que os resultados da estimação tendiam a ser iguais ou superiores a 1m para ambas as coordenadas, quando a quantidade de épocas era inferior a 100. Claro que, como dito anteriormente, foi considerado o que já estava apresentado na literatura. Entretanto, como o que se encontrou na literatura não foi obtido a partir do treinamento de uma rede do tipo MLP, então decidiu-se buscar os valores que mais se adequavam a este modelo de inteligência artificial. Ainda sobre a quantidade de épocas, para treinamentos com épocas acima de 300, o que se percebeu foi piora nos resultados, o que denotou *overtraining*.

Percebe-se que não foram avaliados experimentos sobre o impacto no número de neurônios ou camadas. Isto acontece por que aplicou-se a metodologia MPCA. Como a metodologia já entrega uma arquitetura otimizada, não se faz necessário avaliação quanto a estes parâmetros.

Os experimentos também mostraram aquilo que é explicado na literatura com relação ao erro de deriva do sensor INS (FILYASHKIN; YATSKIVSKY, 2013; NEUSYPIN et al., 2018; OPANASIUK, 2018). Quando foram utilizados os dados do início da trajetória, para o treinamento da MLP, os resultados de teste foram piores. Na abordagem *k-fold*, os dados após o início da trajetória foram utilizados para treinamento, o que refletiu em uma melhora no conjunto de teste. Na Figura 6.5, observou-se que os últimos conjuntos apresentam o erro médio na trajetória maior, isto porque eles usam os dados do final da trajetória para generalização. Consequentemente, torna-se interessante no conjunto de treinamento colocar dados que reflitam o erro de deriva

e que tendam a se mostrar após o início do voo. Por isso, a etapa que utilizou *k-fold* foi importante para encontrar os conjuntos, em que a aprendizagem é melhorada.

Vale lembrar que foram fusionados 12 tipos de dados e que cada dado possui sua parcela de contruição com relação a influência na organização dos dados de treinamento, validação e generalização. Principalmente, sensores como giroscópio e acelerômetro, apresentam alta sensibilidade a qualquer variação na atitude da aeronave durante o voo, o que pode refletir no desempenho durante o treinamento da rede. Contudo, estes sensores não devem ser desprezados, inclusive, na literatura, há resultados que apontam para melhor estimação na fusão quando se combina mais sensores, principalmente os sensores de atitude (CONTE; DOHERTY, 2008; CONTE; DOHERTY, 2009; KIM et al., 2011; PAULINO, 2019; P.; KHANDEKAR, 2019). Ainda sobre sensores de atitude, deve-se ressaltar que sua precisão ou sensibilidade é dependente dos componentes utilizados na fabricação do VANT.

A Tabela 6.6 resume os erros médios para as coordenadas, considerando os experimentos anteriores e seus respectivos métodos de validação cruzada. Para facilitar a leitura da tabela, os erros são apresentados nas colunas de cada experimento e os experimentos são denominados por siglas, como **EX1** para o primeiro experimento, **EX2** para o segundo experimento e assim por diante. A última coluna apresenta a redução ou aumento obtidos e também são apresentados a redução entre os métodos de validação cruzada. Ressalta-se que a redução entre as técnicas hold-out e *k-fold* foi calculada, considerando o conjunto que apresentou o menor erro de estimação para trajetória.

Tabela 6.6 - Erros médios de cada coordenada por experimento.

Abordagem <i>hold-out</i>			
Coordenada	EX1	EX2	Aumento
Latitude	47,57cm	74,70cm	57,03%
Longitude	62,83cm	111,41cm	77,31%
Abordagem <i>k-fold</i>			
Coordenada	EX3	EX4	Redução
Latitude	43,92cm	35,74cm	18,62%
Longitude	59,30cm	53,73cm	9,39%
Redução entre as técnicas <i>hold-out</i> e <i>k-fold</i>			
Coordenada	<i>hold-out</i>	<i>k-fold</i>	Redução
Latitude	47,57cm	35,74cm	24%
Longitude	62,83cm	53,73cm	14%

## 6.2 Avaliação ao aplicar filtragem

Os experimentos anteriores mostraram o impacto na forma como se escolhe os conjuntos de treinamento, validação e generalização. Além disto, algo importante também foi evidenciado quanto a sensibilidade a ruído - percebida claramente por inspeção visual nas imagens.

Na literatura, os estudos abordam a existência de ruído em diversos tipos de dados (CONTE; DOHERTY, 2008; CONTE; DOHERTY, 2009; FRADEN, 2010; KHALEGHI et al., 2013; VALAVANIS; VACHTSEVANOS, 2014; PAULINO, 2019). Ressalta-se ainda que a presença de ruído pode afetar a complexidade do modelo de MLP, o tempo gasto para treinamento e o resultado da predição sobre o conjunto de validação ou novos conjuntos de teste. Por isso, o tratamento de ruídos é um passo importante durante a etapa de seleção dos dados, que acontece antes do treinamento com o objetivo de melhorar a qualidade dos dados e diminuir possíveis efeitos negativos para as fases de validação e generalização.

Na literatura, há fontes que indicam onde aplicar o tratamento de ruídos para dados de voo de uma aeronave do tipo VANT (CONTE; DOHERTY, 2008; CONTE; DOHERTY, 2009; FRADEN, 2010; KHALEGHI et al., 2013; VALAVANIS; VACHTSEVANOS, 2014; PAULINO, 2019). O tratamento de ruídos concentra-se nos dados que afetam diretamente a trajetória do voo, que para este conjunto de dados são a latitude e longitude do INS e as informações obtidas pelo acelerômetro e giroscópio - são os sensores que podem apresentar variações rápidas.

Além da identificação de quais dados tratar, na literatura, também encontra-se duas abordagens para o tratamento de ruídos (PAULINO, 2019). A primeira consiste em identificar partes da trajetória de voo que foram mais afetadas por variações rápidas nas coordenadas, uma vez identificadas o que se faz é remover essas partes do conjunto de treinamento e validação, seguindo um critério estatístico. A Segunda abordagem consiste na aplicação de filtros lineares, como, por exemplo, média móvel.

### 6.2.1 Remoção de ruído

O primeiro experimento com filtragem aplicou a primeira abordagem com relação a tratamento de ruído, ou seja, a remoção de variações rápidas na trajetória de voo. Esta remoção é baseada na análise das variações do gradiente do erro entre as informações obtidas pelos sensores embarcados e as estimativas feitas pelo FK (PAULINO, 2019). Para tanto, foi aplicado um tratamento vetorial nos dados em função

do instante de tempo que a amostra temporal foi obtida.

Este tratamento consiste em calcular a diferença, ou erro em módulo, entre os vetores horizontais medidos pelo sensor INS embarcado e as estimações, após aplicar o FK (Equações 6.1, 6.2 e 6.3). Uma vez que tenha definido o erro, o próximo passo é normalizar estes dados, que é necessária para verificar qual é o desvio padrão do erro visualmente. Sem a normalização o desvio padrão calculado em graus seria muito pequeno e de difícil visualização gráfica.

$$\vec{V}_i = \langle lat_{ins}, lon_{ins} \rangle \quad (6.1)$$

$$\vec{V}_k = \langle lat_{fk}, lon_{fk} \rangle \quad (6.2)$$

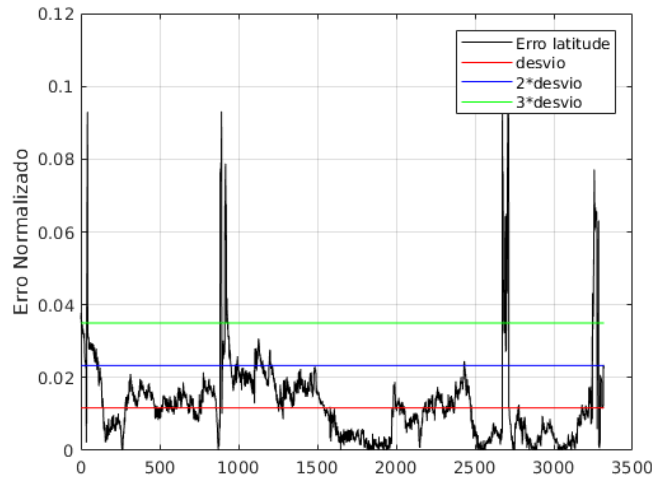
$$\vec{E}_{ik} = \|\vec{V}_i - \vec{V}_k\| \quad (6.3)$$

A Figura 6.6 mostra o erro modular horizontal para as coordenadas. Junto com o erro é apresentado o desvio padrão em três patamares. O primeiro patamar consiste do desvio padrão sem alteração, o segundo é o dobro do desvio padrão e o terceiro é o triplo dele. Desta forma, visualmente, é possível identificar em qual posição na sequência temporal ocorreu um erro maior que algum dos patamares de desvio padrão. Este erro é provocado por uma diferença abrupta entre as medidas e é um forte indicativo de presença de ruídos ou *outliers*.

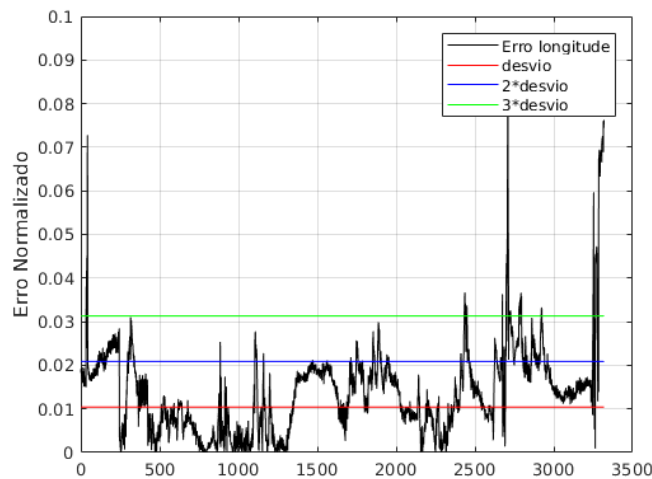
Considerando os instantes de tempo onde o erro ultrapassou o terceiro patamar de desvio, foi possível realizar uma inspeção e encontrar partes ruidosas no conjunto de dados. Estas partes ruidosas foram removidas e assim obteve-se um novo conjunto para experimento. Obviamente a remoção provocou uma redução no total de dados, que passaram de 3.319 para 2.954 amostras temporais. A Figura 6.7 mostra o novo conjunto de dados após a remoção do ruído para ambas as coordenadas. As Figuras 6.8 e 6.9 mostram os intervalos de remoção de ruído.

Uma vez obtido um novo conjunto, com a remoção das partes ruidosas, repetiu-se o processo de treinamento da rede neural. Dado que nos experimentos da Sessão 6.1.2 mostrou-se o impacto da forma de dividir os dados, foi possível encontrar a divisão que apresentou o menor erro médio na trajetória, esta divisão foi selecionada para

Figura 6.6 - Erro modular horizontal.



(a)



(b)

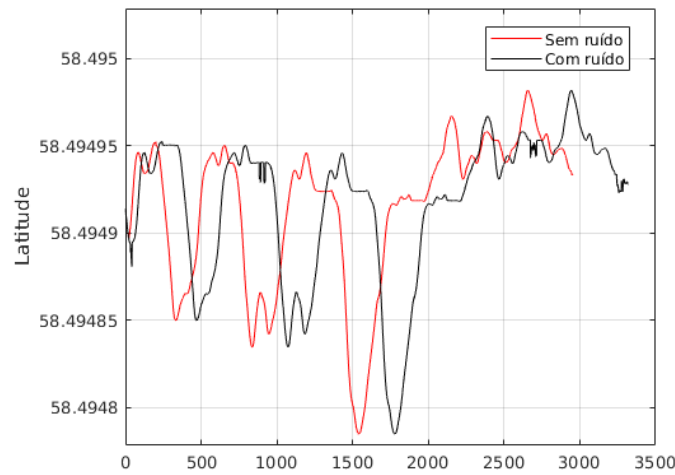
(a) Valores para a coordenada de latitude. (b) Valores para a coordenada de longitude.

Fonte: Produção do autor.

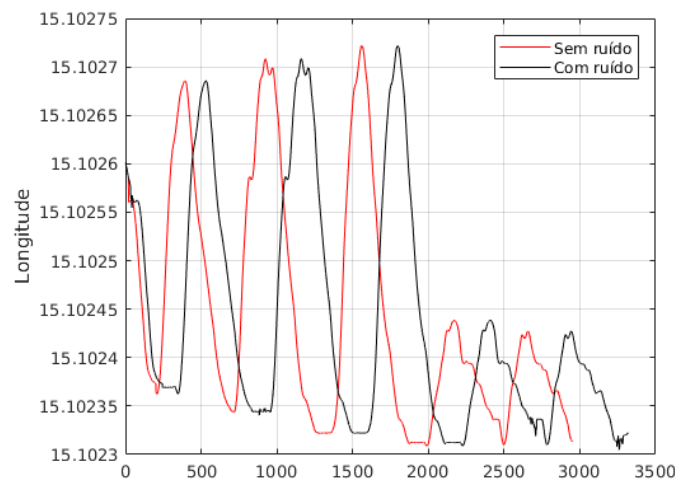
a avaliação nas etapas de remoção e tratamento de ruído.

Neste experimento, o resultado da generalização, após a remoção das partes ruidosas, apresentou melhora nas estimações. Os erros médios para as coordenadas de latitude e longitude foram, respectivamente, 30,14cm e 33,27cm. Considerando o resultado para o mesmo conjunto de dados, antes da remoção das partes ruidosas, houve redução no erro médio de estimação da latitude de aproximadamente 6cm. Já para

Figura 6.7 - Novo conjunto de dados, após a remoção de ruído.



(a)



(b)

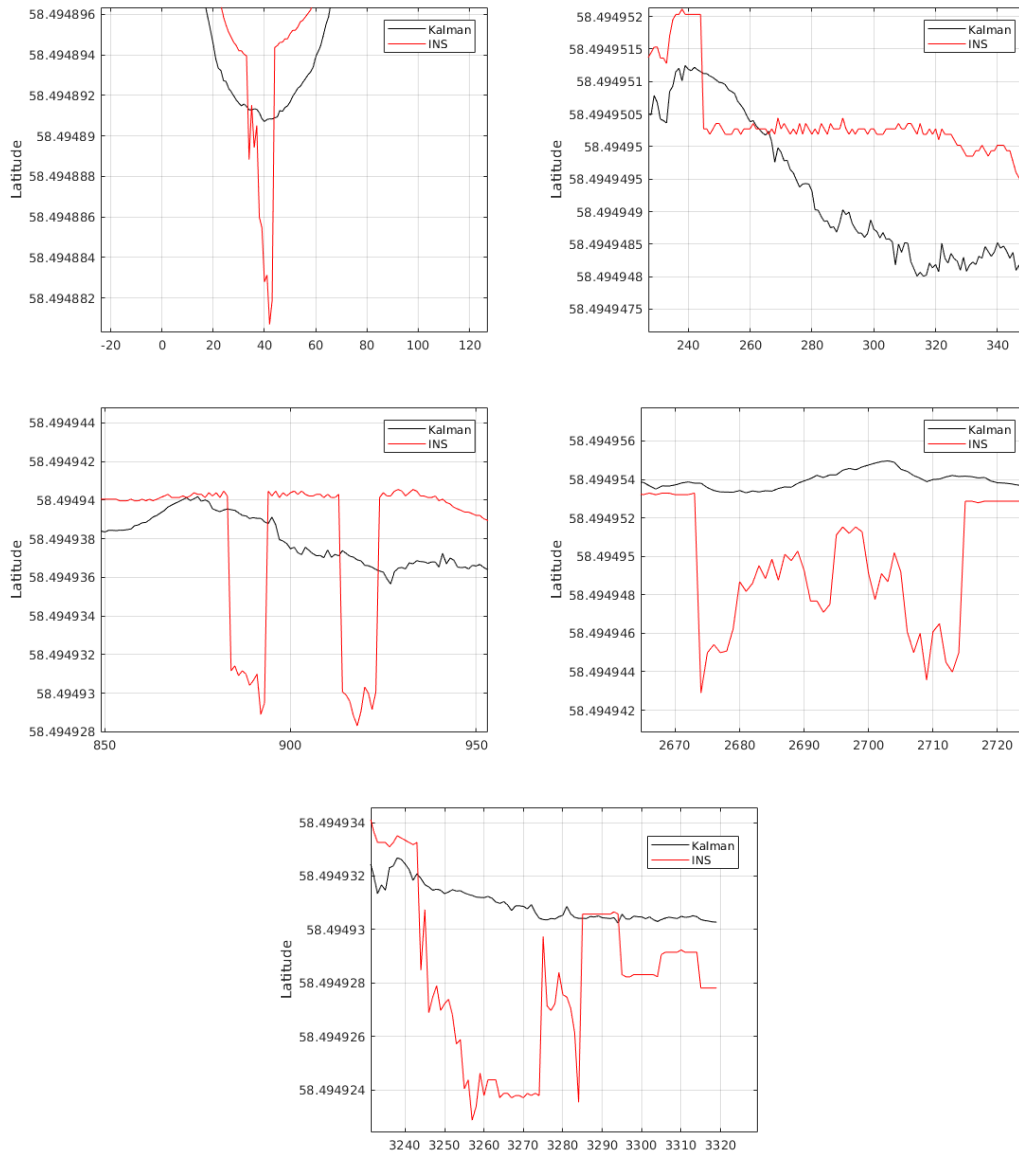
(a) Valores para a coordenada de latitude. (b) Valores para a coordenada de longitude.

Fonte: Produção do autor.

a longitude, a redução no erro médio de estimação foi melhor, de aproximadamente 20cm.

Uma vez que houve redução nos erros médios das coordenadas, o erro médio na trajetória também foi impactado e o valor obtido foi de 49,19cm. Quando comparado com o erro médio na trajetória, no primeiro experimento da Sessão 6.1.1, a redução obtida foi de 43%. A Tabela 6.7 mostra as incertezas para este experimento,

Figura 6.8 - Intervalos de remoção de ruído para latitude.



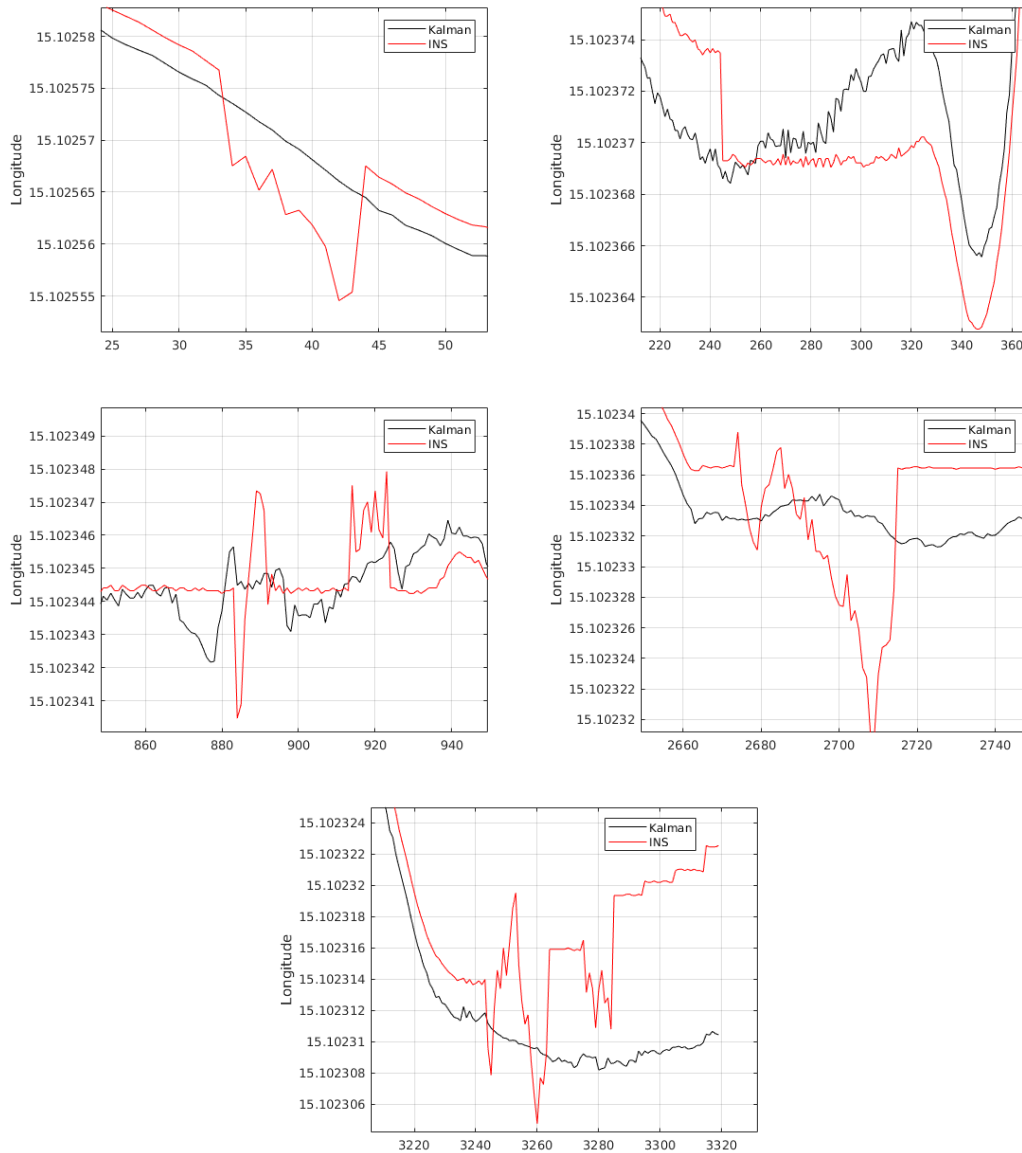
Fonte: Produção do autor.

as reduções apresentadas são calculadas a partir do primeiro experimento da Sessão 6.1.1. Percebe-se o impacto maior obtido no desvio e variância, isto corrobora com a suspeita de que a presença de ruído interfere no intervalo de confiança das estimações.

Neste experimento, o treinamento foi aplicado ao conjunto de dados que apresentou o menor erro de generalização durante a investigação com a técnica *k-fold*. Então,



Figura 6.9 - Intervalos de remoção de ruído para longitude.



Fonte: Produção do autor.

torna-se interessante apresentar a redução nas incertezas comparadas ao mesmo conjunto sem a remoção das partes ruidosas. Isto é mostrado na Tabela 6.8, ressaltando que os valores apresentados são os erros médios. De certo modo, seria injusto a comparação apenas com conjuntos de dados que não foram avaliados quanto a organização e divisão dos dados de treinamento, validação e generalização.

Para facilitar a leitura, utilizou-se a mesma denominação aplicada anteriormente,

Tabela 6.7 - Incertezas em centímetros para cada coordenada, após remoção das partes ruidosas - quinto experimento.

Latitude		
Tipo de incerteza	Valor	Redução na incerteza
Erro médio	30,14cm	36,64%
Desvio	22,59cm	50,61%
Variância	510,34cm	75,61%
Longitude		
Tipo de incerteza	Valor	Redução na incerteza
Erro médio	33,27cm	47,04%
Desvio	27,49cm	44,41%
Variância	756,05cm	69,10%

como na Tabela 6.6, onde os experimentos foram denominados de acordo com sua sequência. Como este é o quinto experimento, na Tabela 6.8 foram comparados o **EX4** com o **EX5**. Novamente, cabe o destaque na redução do erro médio para o intervalo de confiança, em que o impacto desta redução influenciou positivamente o erro médio na estimação da trajetória, onde houve melhora de 29,13%. As estimações para as coordenadas também foram melhoradas, o destaque foi para longitude, onde ocorreu uma melhora na estimação de 37%, com grande reflexo sobre o desvio e a variância do erro médio. Mais uma vez, evidenciou-se que a presença de ruído ou *outliers* causa um impacto negativo no aprendizado da rede neural, o que se reflete sobre o cálculo das incertezas.

Tabela 6.8 - Comparação entre EX4 e EX5.

Coordenada	EX4	EX5	Redução
Erro médio			
Latitude	35,74cm	30,14cm	15,66%
Longitude	53,37cm	33,27cm	37,66%
Desvios			
Latitude	24,62cm	22,59cm	8,24%
Longitude	43,26cm	27,49cm	36,45%
Variância			
Latitude	606,61cm	510,34cm	15,87%
Longitude	1.871,60cm	756,05cm	59,60%
Trajetória			
-	69,41cm	49,19cm	29,13%

## 6.2.2 Aplicação de filtros lineares

Pela Sessão 6.2.1, ficou claro o impacto negativo que a presença de ruído ou de outliers pode provocar. Porém, apesar da remoção das partes ruidosas, a forma aplicada para tratar o ruído não afetou a série temporal como um todo, ou seja, algumas partes foram removidas, mas o restante dos dados continuou original.

Como mencionado anteriormente, na literatura, é descrita a presença de ruído para estes tipos de dados. Além disto, a própria eletrônica de alguns sensores torna as aquisições propícias para incorporar ruídos (FRADEN, 2010). Por exemplo, entre os sensores que são propícios em registrar variações rápidas, que podem ser ruidosas, existe o giroscópio. Isto fica mais evidenciado quando é feita uma inspeção visual na série temporal obtida por este sensor, como mostra a Figura 6.10.

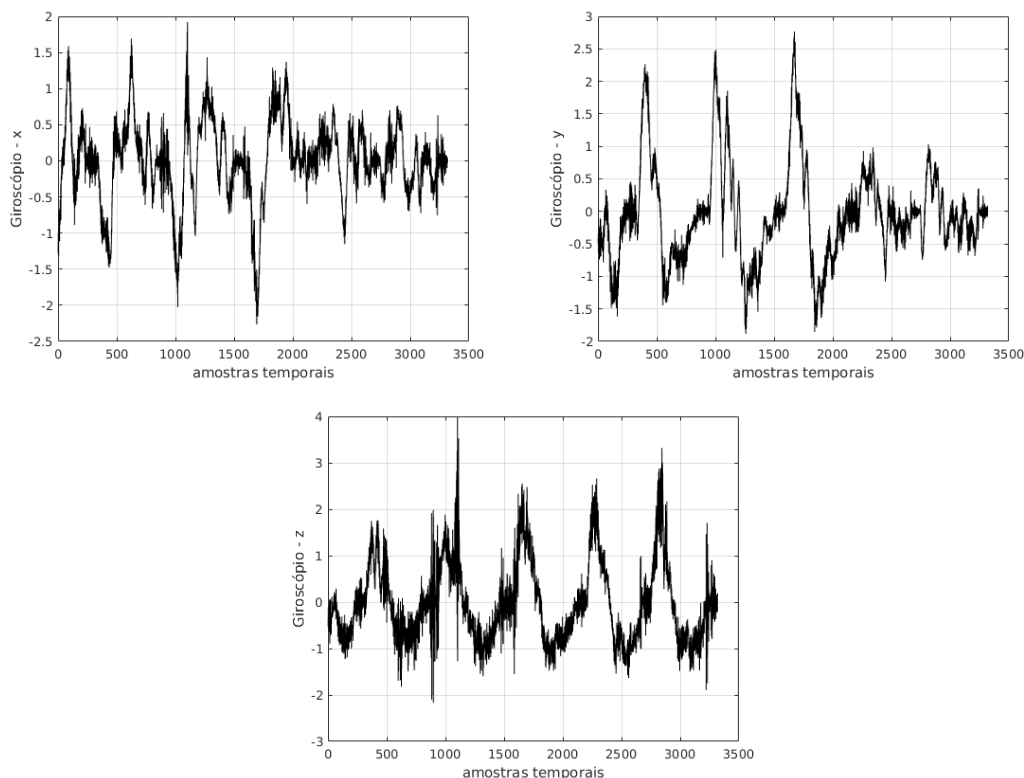
Assim como as informações do giroscópio, os valores das coordenadas de latitude e longitude obtidos pelos INS também apresentam variações, caracterizando a presença de ruído. Estas variações são pequenas em amplitude e, por isso, não foram detectadas na primeira abordagem para remoção de ruído. Pode-se dizer que também não faria sentido remover estas variações do modo anterior, pois resultaria na degradação da série temporal.

Uma vez que ainda existe a presença de ruído, a solução utilizada para suavizar as variações rápidas foi a aplicação de um filtro linear. Contudo, na literatura, é possível encontrar vários tipos de filtros e investigar todas as possibilidades, porém seria uma tarefa exaustiva. Então, optou-se por concentrar em alguns, já conhecidos e aplicados no contexto da navegação de VANT (CONTE; DOHERTY, 2008; CONTE; DOHERTY, 2009; KIM et al., 2011; KHALEGHI et al., 2013; BRAGA et al., 2015; BRAGA et al., 2016; BRAGA, 2018, 2018; PAULINO, 2019). Foram avaliados os seguintes tipos de filtro:

- Filtro média móvel;
- Filtro mediana móvel;
- Filtro gaussiano;
- Filtro Savitzky-Golay.

Todos estes filtros utilizam uma abordagem baseada em janela deslizante, também chamada de janela móvel, em que o tamanho da janela define os intervalos utilizados

Figura 6.10 - Valores obtidos pelo giroscópio para cada eixo.



Fonte: Produção do autor.

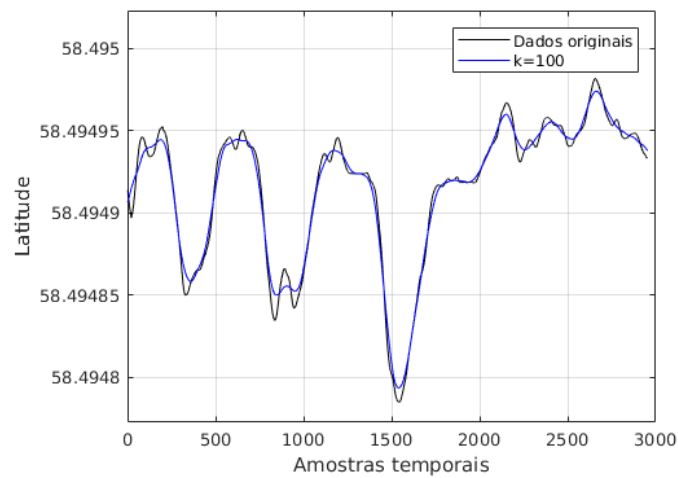
para cálculo da suavização. Então, torna-se importante escolher qual será o comprimento destes intervalos, uma vez que isto impacta diretamente na intensidade da filtragem. Em outras palavras, caso o tamanho escolhido para janela móvel seja grande demais, a filtragem passaria a degradar a série temporal, o que provocaria aumento no erro das estimações.

Para facilitar a explicação de como foi selecionado o tamanho da janela móvel, denominou-se este tamanho pela letra **k**. Foi tomado como base o primeiro filtro da lista anteriormente citada, que seria o filtro média móvel com o menor tamanho possível, que seria de  $k = 2$ , por exemplo. Entende-se, então, que  $k = 2$  representa a menor intensidade de filtragem e a medida que aumenta o valor de  $k$ , aumentando a intensidade da suavização provocada pelo filtro. Para ilustrar o impacto no aumento do valor de

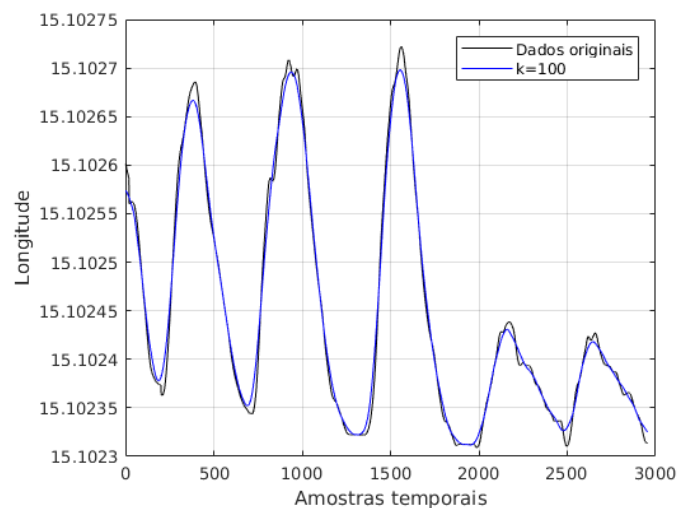
Percebe-se pela Figura 6.11, que quando o valor de  $k$  está alto, ele provoca uma alteração no formato da curva, que define os dados originais. Portanto, cabe calcular

a diferença, o erro médio, entre os dados originais e os dados filtrados. Esta diferença para latitude foi de 44,02cm e para longitude foi de 44,29cm. Claramente, também existe uma diferença relacionada a trajetória, que neste caso foi de 68,53cm. Apenas para retificar, estas diferenças não são provenientes do resultado da ativação da rede neural, elas correspondem a diferença entre os dados originais e os dados filtrados, ou seja, calculou-se quanto seria a degradação em centímetros após aplicar o filtro média móvel com  $k = 100$ .

Figura 6.11 - Suavização do filtro média móvel para  $k = 100$ .



(a)



(b)

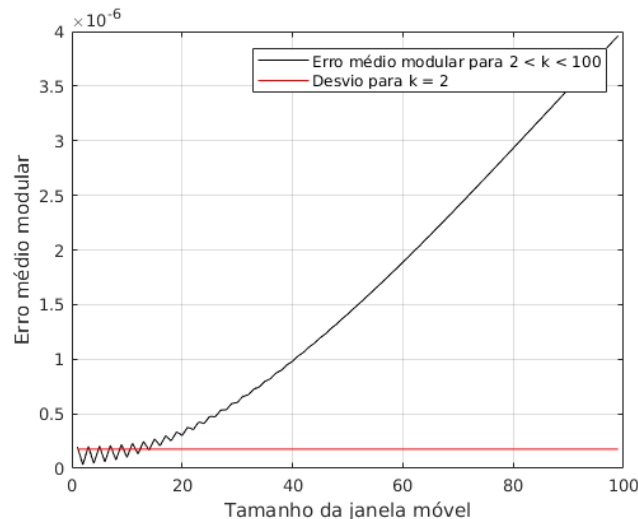
(a) Valores para coordenada latitude. (b) Valores para a coordenada longitude.

Fonte: Produção do autor.

Uma vez evidenciado a degradação, quando se aumenta o valor de  $k$ , torna-se necessário definir um valor para os testes com filtragem. Para  $k = 2$ , que é a menor intensidade possível para filtragem, existe um erro associado e um desvio, com relação a trajetória. Desta forma foi escolhido o maior valor para  $k > 2$ , em que o erro médio na trajetória não ultrapassa o valor de desvio quando  $k = 2$ . Assim, tentou-se garantir a maior intensidade possível de filtragem, com o mínimo de degradação em relação aos valores originais das séries temporais de latitude, longitude e demais sensores.

Na Figura 6.12, é mostrado o erro médio modular na trajetória, em graus, para  $k$  variando de 2 até 100. Ressalta-se que o módulo do erro médio foi aplicado para que se pudesse, de maneira visual, verificar a curva crescente do erro. Além do erro médio modular, é mostrado na curva de cor vermelha o desvio quando  $k = 2$ . A partir disto, foi possível selecionar o valor  $k = 15$ , que foi o valor que se encaixou na métrica aplicada para escolha do tamanho da janela móvel. Pode-se perceber pela Figura 6.12 que existem valores de  $k$  menores do que 15, que ultrapassaram o patamar de desvio quando  $k = 2$ , o que foi percebido para os valores de  $k$  é par. Já os valores de  $k$  ímpar, se mantiveram abaixo do patamar de desvio.

Figura 6.12 - Seleção do tamanho da janela móvel, para o filtro média móvel.



Erro médio calculado sobre a trajetória.

Fonte: Produção do autor.

A análise sobre o valor de  $k$  fica demonstrada na Figura 6.12, que foi realizada para

o filtro média móvel, mas este não foi o único filtro testado. Desta forma, a mesma métrica foi aplicada aos demais filtros da listagem anteriormente mencionada. Os valores do tamanho da janela móvel ou ordem do filtro junto com as diferenças nas coordenadas e trajetória para cada filtro, são apresentados na Tabela 6.9. As diferenças mostradas na Tabela 6.9 correspondem ao erro médio. Ressalta-se que as diferenças nas coordenadas e trajetória foram calculadas em relação aos dados originais obtidos pelo sensor INS.

Tabela 6.9 - Tamanho da janela móvel e diferenças para cada filtro.

		Latitude	Longitude
Filtro	k	diferença	diferença
Média móvel	15	1,85cm	2,12cm
Mediana móvel	33	2,19cm	2,98cm
Gaussiano	25	2,14cm	2,45cm
Savitzky-Golay	45	1,83cm	2,52cm
Trajetória			
Filtro	k	diferença	
Média móvel	15	3,12cm	
Mediana móvel	33	3,93cm	
Gaussiano	25	3,59cm	
Savitzky-Golay	45	3,41cm	

Percebe-se pela Tabela 6.9 que o erro médio na trajetória manteve-se próximo a 3cm, para todos os filtros. Desta forma, foi possível realizar uma comparação mais justa, afim de investigar qual filtragem pode melhorar o desempenho para a estimação da posição. Cabe ressaltar que aconteceu da mesma forma, como no experimento anterior. Isto fica explicitado na Sessão 6.2.1, onde demonstra-se como foi aplicado a remoção de partes ruidosas. Nesta etapa, a divisão dos dados entre treinamento, validação e generalização manteve-se da mesma forma, sendo que a filtragem não foi aplicada sobre o conjunto de generalização.

Após a filtragem dos dados de treinamento, um novo experimento foi realizado. Considerando o erro médio na trajetória, o filtro que apresentou o melhor resultado foi o Savitzky-Golay, com um erro médio de 57,78cm. O pior resultado foi obtido com o filtro mediana móvel, com um erro médio de 62,49cm. Apesar do filtro mediana móvel ter obtido o pior resultado, a diferença entre os erros foi pequena, próximo de 4cm. Isto é reflexo da intensidade da filtragem, que provocou diferenças próximas entre os dados originais e os dados filtrados.

Com relação as coordenadas, o filtro Savitzky-Golay também obteve os melhores resultados, o que já era esperado, pois, como dito anteriormente, foi o que obteve o melhor resultado na trajetória. Para o filtro Savitzky-Golay o erro médio na estimação da latitude foi de 31,22cm e para estimação da longitude foi 43,56cm. As piores estimações para as coordenadas foram obtidas pelos filtros mediana móvel e gaussiano. Especificamente o filtro gaussiano obteve um erro médio na estimação latitude de 37,09cm, já o filtro mediana móvel obteve um erro de estimação na longitude de 46,61cm. A Tabela 6.10 mostra as incertezas para cada coordenada, incluindo o erro médio na trajetória, para cada filtro.

Tabela 6.10 - Incertezas em centímetros para cada coordenada, após aplicação de filtros lineares - sexto experimento.

Filtro	Latitude		Longitude	
	Tipo de incerteza	Valor	Tipo de incerteza	Valor
Média móvel	Erro médio	32,97cm	Erro médio	44,47cm
	Desvio	24,66cm	Desvio	33,98cm
	Variância	608,51cm	Variância	1.154,77cm
Mediana móvel	Erro médio	33,60cm	Erro médio	46,61cm
	Desvio	28,73cm	Desvio	27,80cm
	Variância	825,41cm	Variância	773,22cm
Gaussiano	Erro médio	37,09cm	Erro médio	43,75cm
	Desvio	20,85cm	Desvio	31,11cm
	Variância	435,06cm	Variância	967,88cm
Savitzky-Golay	Erro médio	31,22cm	Erro médio	43,56cm
	Desvio	18,57cm	Desvio	29,18cm
	Variância	345,05cm	Variância	851,98cm
Erro médio na trajetória				
Filtro		Valor		
Média móvel		59,56cm		
Mediana móvel		62,49cm		
Gaussiano		61,74cm		
Savitzky-Golay		57,78cm		

Ressalta-se que a aprendizagem que utilizou dados filtrados pelo Savitzky-Golay obteve o melhor resultado entre os demais filtros lineares avaliados. Cabe, então, comparar seu resultado de estimação com os resultados obtidos nos experimentos anteriores, especificamente o **EX4** e o **EX5**.

Com relação ao **EX4**, os resultados após a aplicação do filtro linear foram unanimemente melhores, houve uma redução nos erros médios da estimação em ambas



as coordenadas. Aponta-se que o erro médio para estimação da latitude reduziu em 12,64% e para estimação de longitude a redução do erro foi de 18,38%. Claramente, isto impactou o erro médio na trajetória, que reduziu 16,75%.

Com relação ao **EX5**, os resultados não foram melhores, o erro médio para estimação da latitude aumentou em 3,58% e para estimação da longitude ocorreu um aumento de 30,92%. O erro médio na trajetória aumentou 17,46%. Contudo, ocorreu um ponto positivo, que foi a redução no intervalo de confiança da estimação para latitude que melhorou em 32,38%. A Tabela 6.11 mostra os valores para comparação entre os experimentos. Na coluna diferença, os valores negativos representam redução e os positivos aumento. Como este foi o sexto experimento, na Tabela 6.11, seus valores aparecem na coluna denominada **EX6**.

Tabela 6.11 - Comparação com EX4 e EX5.

Coordenada	Comparação com EX4			Comparação com EX5		
	EX4	EX6	Diferença	EX5	EX6	Diferença
Erro médio						
Latitude	35,74cm	31,22cm	-12,64%	30,14cm	31,22cm	3,58%
Longitude	53,37cm	43,56cm	-18,38%	33,27cm	43,56cm	30,92%
Desvio						
Latitude	24,62cm	18,57cm	-24,57%	22,59cm	18,57cm	-17,79%
Longitude	43,26cm	29,18cm	-32,54%	27,49cm	29,18cm	6,14%
Variância						
Latitude	606,61cm	345,05cm	-43,11%	510,34cm	345,05cm	-32,38%
Longitude	1.871,60cm	851,98cm	-54,47%	756,05cm	851,98cm	12,68%
Erro médio na trajetória						
-	69,41cm	57,78cm	-16,75%	49,19cm	57,78cm	17,46%

### 6.2.3 Considerações sobre a avaliação ao aplicar filtragem

Os ruídos ou variações abruptas nas medidas obtidas durante o voo podem ser provocadas por diferentes situações, que vão desde rajadas de vento, que deslocam ligeiramente o VANT de posição, até a qualidade dos sensores embarcadas (FRADEN, 2010; VALAVANIS; VACHTSEVANOS, 2014). Por isso, controlar ou prever situações que vão gerar medidas ruidosas é difícil. Contudo, a avaliação mostrou que usar dados no conjunto de treinamento, com variações abruptas, não é adequado à medida que provoca um impacto negativo nas estimações. Desta forma, conclui-se que para o treinamento da MLP é importante a remoção das variações abruptas.

Apesar de que remover as variações abruptas provocaram melhora na estimação da posição, o mesmo não ocorreu com a aplicação de filtros lineares. Neste caso, vale lembrar que as variações nas medidas de latitude e longitude para o voo estudado acontecem nas últimas casas decimais, a partir da quinta casa decimal. Então, a suavização da filtragem linear altera a informação na mantissa dos dados, o que provoca piora no treinamento e, como consequência, piora nas estimações. Desta forma, entende-se que somente variações abruptas devem ser removidas do conjunto de treinamento.

Outro ponto que pode ser destacado, com relação a aplicação de filtros lineares, foi o cuidado na seleção do tamanho da janela móvel, de tal forma que a degradação dos dados originais foi mínima. Mesmo com a degradação mínima dos dados originais, percebeu-se piora na estimação. Portanto, aumentar o tamanho da janela móvel não irá melhorar o treinamento e tampouco os resultados de estimação.

### **6.3 Combinação dos sensores de atitude**

Como dito anteriormente, o custo de fabricação e manutenção de um VANT é menor quando comparado a aeronaves tripuladas. Isto, inclusive, é um dos motivos de VANT ser uma alternativa interessante quando não é possível usar aeronaves tradicionais (FUHRMANN; HOROWITZ, 2017). Contudo, sabe-se que o desempenho dos sensores e demais dispositivos, que são embarcados no VANT, pode depender da qualidade dos materiais eletrônicos e componentes utilizados na fabricação da aeronave (FRADEN, 2010; VALAVANIS; VACHTSEVANOS, 2014). Por isso, VANTs que possuem maior capacidade de carga, maior tempo de voo e sensores com melhor precisão custam mais do que versões de VANTs mais simples.

Na intenção de manter o equilíbrio entre custo de fabricação e desempenho, alguns fabricantes acabam incorporando à malha de controle do VANT os sensores com alto desempenho, juntamente com sensores de baixo desempenho. Dado que sensores com baixo desempenho possuem menores precisões, pode-se inferir que eles prejudicam a precisão nas estimações da posição, quando utilizados na fusão de dados. Assim, torna-se interessante avaliar o desempenho da fusão de dados a partir da escolha ou combinação de alguns sensores ou ainda da combinação de determinados dados.

#### **6.3.1 Comparação entre acelerômetro e giroscópio**

Para um voo de VANT, é importante saber o posicionamento da aeronave no espaço. O posicionamento no espaço consiste em informações como altura do voo, inclinação

do corpo da aeronave, direção e sentido da navegação e, por último, aceleração. A partir deste posicionamento, é possível escolher as ações a serem efetuadas durante a navegação.

As informações do posicionamento no espaço são entregues a controladora de voo vindas do sensor INS, especificamente dos componentes altímetro, acelerômetro e giroscópio. Cada um desses componentes pode ser entendido como um sensor em separado, uma vez que cada um entrega um tipo de dado específico. As informações obtidas destes sensores denotam o estado e a tendência de movimentação do VANT no espaço. Na literatura, eles são chamados de sensores de atitude (CONTE; DOHERTY, 2008; CONTE; DOHERTY, 2009; LIU; ZHANG, 2012; BAO et al., 2017; PAULINO, 2019).

Na intenção de avaliar o desempenho da fusão de dados, um outro experimento foi realizado sobre os dados do voo feito na Suécia a partir da combinação dos sensores de atitude. Neste experimento, os dados obtidos dos sensores acelerômetro e giroscópio foram alternados de dentro do conjunto de dados usados anteriormente. Portanto, foram montados dois conjuntos de dados, em que o primeiro conjunto tinha todos os dados usados nos experimentos anteriores, menos os dados do sensor acelerômetro. Já o segundo conjunto continha todos os dados usados nos experimentos anteriores, menos os dados do sensor giroscópio. A partir destes novos conjuntos gerados, novamente foi aplicada a fusão de dados sobre cada conjunto.

É importante mencionar que neste experimento o sensor altímetro não foi alternado do conjunto de dados, ou seja, não foi avaliada a fusão com a ausência deste sensor. As informações do sensor altímetro estão presentes nos dois conjuntos formados. Isto porque a altura do voo feito na Suécia foi constante (CONTE; DOHERTY, 2009). Desta forma entende-se que as informações do sensor de altitude não impactariam de modo significativo na fusão de dados, se fossem removidos, pois houve pouca variação em suas medições.

O resultado da fusão de dados aplicada ao primeiro conjunto, que seria o conjunto do acelerômetro, é mostrado na Tabela 6.12. Na Tabela 6.12, também é mostrada a comparação com o experimento **EX5**, que foi o melhor resultado durante a avaliação da filtragem. Como este foi o sétimo experimento sobre os dados do WITAS, as incertezas da fusão de dados são mostradas na coluna **EX7** na Tabela 6.12.

As incertezas da fusão de dados para o primeiro conjunto foram melhoradas. Houve uma redução de 13% no erro médio para estimação da coordenada latitude e uma

redução de 5% para estimação da coordenada longitude. As incertezas que denotam intervalo de confiança também foram melhoradas, com destaque para a variância do erro médio para coordenada longitude que teve 15% de redução. O erro médio para estimação da trajetória foi reduzido em 8%.

Tabela 6.12 - Comparação com EX7 e EX5.

Coordenada	EX5	EX7	Redução
Erro médio			
Latitude	30,14cm	26,00cm	13,73%
Longitude	33,27cm	31,53cm	5,22%
Desvio			
Latitude	22,59cm	21,15cm	6,37%
Longitude	27,49cm	25,21cm	8,29%
Variância			
Latitude	510,34cm	447,63cm	12,28%
Longitude	756,05cm	636,02cm	15,87%
Erro médio na trajetória			
-	49,19cm	44,87cm	8,78%

O resultado da fusão de dados aplicada ao segundo conjunto, que é o conjunto do giroscópio, é mostrado na Tabela 6.13. Como este foi o oitavo experimento sobre os dados do WITAS, as incertezas da fusão de dados são mostradas na coluna **EX8** na Tabela 6.12. As incertezas deste experimento também foram comparadas com o **EX5**, semelhante a comparação na Tabela 6.12.

De maneira geral, neste experimento, também houve melhora nas incertezas. Contudo, o erro médio de estimação para a coordenada de longitude foi piorado em 4%. Anteriormente, no experimento **EX5**, o erro médio para estimação da coordenada longitude foi de 33cm. Porém, para o conjunto do giroscópio este erro foi para 34cm, demarcando uma pequena piora. Devido a esta pequena piora, na Tabela 6.13, os valores que representam diminuição na incerteza estão negativos, já o valor que representa aumento na incerteza está positivo. A coluna que mostra a comparação entre os experimentos é denominada diferença.

Para as demais incertezas houve redução, com destaque para a variância do erro médio na estimação da coordenada longitude que reduziu em 18%. O erro médio para estimação da coordenada latitude reduziu em 12%. O erro médio na estimação da posição para a trajetória teve uma pequena redução de 2%.

Tabela 6.13 - Comparação com EX8 e EX5.

Coordenada	EX5	EX8	Diferença
Erro médio			
Latitude	30,14cm	26,43cm	-12,30%
Longitude	33,27cm	34,66cm	4,17%
Desvio			
Latitude	22,59cm	21,73cm	-3,80%
Longitude	27,49cm	24,85cm	-9,60%
Variância			
Latitude	510,34cm	472,59cm	-7,39%
Longitude	756,05cm	617,63cm	-18,30%
Erro médio na trajetória			
-	49,19cm	47,93cm	-2,56%

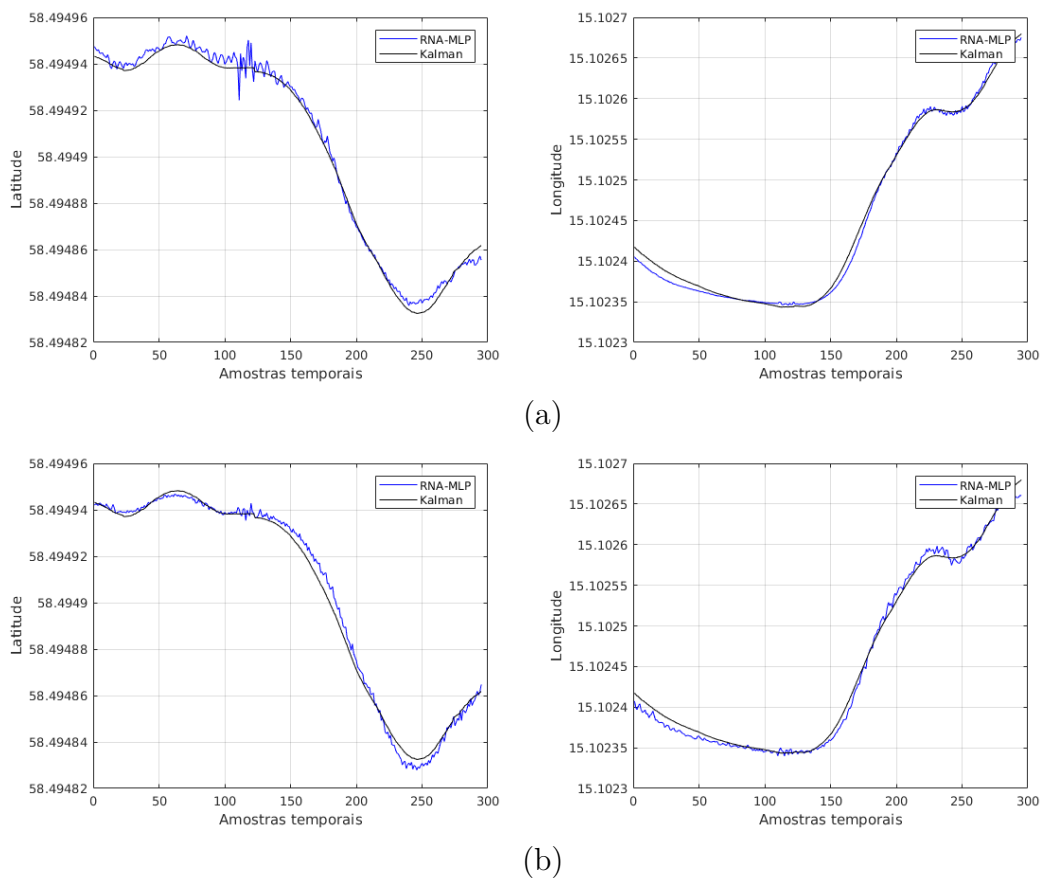
### 6.3.2 Considerações sobre a combinação dos sensores de atitude

Os resultados mostrados nas Tabelas 6.12 e 6.13 apresentaram valores próximos para os erros médios de estimação para cada coordenada e também para o erro médio da estimação na trajetória. Portanto, para a aeronave utilizada neste experimento, conclui-se que usar somente os dados do sensor acelerômetro ou somente os dados do sensor giroscópio, juntamente com os demais dados, produz o mesmo resultado para fusão de dados.

Contudo, se fosse necessário priorizar um dos dois sensores de atitude seria melhor utilizar o sensor acelerômetro, pois na Tabelas 6.12 o erro médio da estimação da trajetória foi de 44cm, 5cm menor que o erro médio da estimação na trajetória mostrado na Tabelas 6.13. Além disso, quando se pensa na atitude da aeronave, ou seja, no estado e tendência de movimento do VANT, o sensor acelerômetro entrega como informação a aceleração do corpo da aeronave em relação a sua inércia, isto melhora a estimativa de deslocamento feita pelo fusor.

Uma vez que se tem o resultado da fusão de dados, para os dois conjuntos, torna-se interessante mostrar, graficamente, o resultado das estimações para cada coordenada. A Figura 6.13 mostra as estimações de cada coordenada, para cada um dos conjuntos. Percebe-se que as estimações se aproximam bastante do FK e que há pouca diferença entre os resultados de cada conjunto.

Figura 6.13 - Resultados da fusão de dados para os conjuntos com acelerômetro ou giroscópio.



(a) Resultado da estimação para ambas as coordenadas, considerando os dados do sensor acelerômetro. (b) Resultado da estimação para ambas as coordenadas, considerando os dados do sensor giroscópio.

Fonte: Produção do autor.

## 7 SEGUNDA FASE DE EXPERIMENTOS - DADOS DO IEAV

Os resultados da Capítulo 6 foram animadores. Isto estimulou uma segunda fase de experimentos com voo feito no Brasil. Realizar testes na América Latina é interessante uma vez que nesta região há possibilidade de ocorrência dos fenômenos AMAS e CI (PAULA et al., 2007; BALAN et al., 2018; BARROS et al., 2018). Nesta segunda fase de experimentos buscou-se verificar duas questões.

A primeira questão verificada foi o erro associado à navegação por visão computacional. Como descrito anteriormente a fusão de dados processa informações do sensor INS juntamente com informações de um sistema que estima a posição do VANT por processamento de imagens e uma das formas de estimação da posição por processamento de imagens é chamada de visão computacional (BRAGA et al., 2016; BRAGA, 2018). Deste modo buscou-se calcular o erro na estimação da posição para à navegação por visão computacional.

A segunda questão foi verificar a independência da metodologia proposta em relação ao tipo de aeronave ou componentes utilizados na fabricação dos sensores. Cabe ratificar que o VANT utilizado nesta fase não foi o mesmo utilizado no voo na Suécia, tampouco, esta aeronave possui sensores iguais ou com a mesma precisão. Desta forma, pôde-se mostrar que a metodologia não é dependente das características do VANT, ou seja, a metodologia não depende da dinâmica da aeronave. Isto é importante dado que nos experimentos realizados pelo grupo de pesquisa do WITAS, o FK aplicado como fusor de dados é dependente da dinâmica da aeronave (CONTE; DOHERTY, 2008; CONTE; DOHERTY, 2009), ou seja, no caso estudado na Suécia, não era permitido trocar a aeronave, a troca do VANT iria inutilizar o FK aplicado na fusão de dados.

Este capítulo está dividido nas seguintes fases:

- Erro associado à navegação por visão computacional;
- Inclusão dos dados inerciais;
- Desenvolvimento de um coprocessador em *hardware* dedicado.

É importante informar que as incertezas mostradas neste capítulo foram calculadas sobre a diferença entre o que foi obtido pela MLP autoconfigurada e o que foi obtido por uma fonte de referência. Neste capítulo a fonte de referência é um GPS-RTK, mais precisamente, a trajetória medida por este equipamento - que é de alta precisão.

O GPS-RTK usado foi embarcado no VANT utilizado nos experimentos. Também torna-se importante informar que o erro médio para as medidas da trajetória associadas ao INS, embarcado, é de 200cm, quando compara-se estas medidas com as medidas obtidas pelo GPS-RTK (PAULINO, 2019; DJI, 2020a).

### 7.1 Erro associado à navegação por visão computacional

Nesta etapa, foram realizados experimentos com o objetivo de avaliar e refinar a metodologia proposta para navegação autônoma utilizando visão computacional e fusão de dados. O primeiro experimento buscou identificar se existe um erro associado à estimativa da primeira posição realizada por visão computacional. Cabe informar que todas as imagens usadas nos experimentos desta Sessão, foram obtidas por um sensor óptico.

Como descrito, anteriormente, existem diferentes métodos para estimar a posição de um VANT por processamento imagens (CONTE; DOHERTY, 2009; SILVA FILHO, 2016; BRAGA et al., 2018; ROOS, 2018; DOMICIAN, 2018; AL-KAFF et al., 2018; REBOUÇAS, 2019). Nos experimentos desta Sessão foi aplicado a correlação cruzada entre duas imagens (SILVA, 2016; BRAGA, 2018). O processo de estimação da posição foi aplicado após a extração das bordas de uma imagem georreferenciada e das imagens coletadas durante a navegação do VANT. A Figura 7.1 mostra a imagem georreferenciada utilizada nos experimentos.

Figura 7.1 - Imagem georreferenciada, usada na estimação por visão computacional.



Fonte: Produção do autor.



Sobre a imagem mostrada na Figura 7.1 cabe evidenciar que ela não é uma imagem de satélite. Esta imagem foi capturada por um VANT do modelo *DJI Phantom 4* (DJI, 2020b). Este VANT possui uma câmera embarcada e integrada com seu sistema de navegação. A câmera embarcada é do modelo *FC-330*, da fabricante SONY (LIMA, 2018). Possui resolução de 12 *megapixels* e distância focal nominal de 3,61mm (LIMA, 2018). Pode-se entender a distância focal, de maneira simplificada, como a medida em milímetros que define qual será a aproximação do objeto a ser fotografado em relação a imagem final. Também conhecida como comprimento focal, ela representa a capacidade da lente de expandir a captura da imagem em um determinado ponto, mesmo que distante (LIMA, 2018).

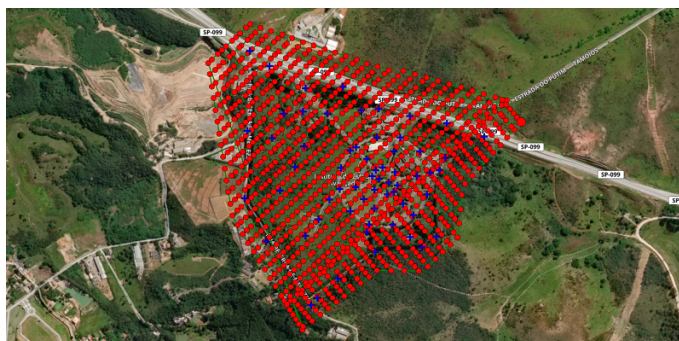
Cabe evidenciar também que esta aeronave foi pilotada, utilizando-se um controle em solo. Portanto não aconteceu um voo autônomo para capturar a imagem georreferenciada. Além disso, a trajetória do VANT utilizado para capturar a imagem georreferenciada, precisou ser previamente planejada (LIMA, 2018). Para o planejamento de voo utilizou-se um *software* denominado *Pix4D Capture* (PIX4D, 2020). Este *software* é especializado em gerar plano de voo para obtenção de fotografias com superposição longitudinal e lateral.

A imagem mostrada na Figura 7.1, foi construída a partir da sobreposição de imagens capturadas durante a navegação. Para isto, foi necessário realizar uma cobertura fotográfica da área sobrevoada.

Para realizar a cobertura fotográfica de área de interesse foi necessário delimitar no *software Pix4D Capture* a extensão da área sobrevoada e informar a altura de voo em relação ao solo, com isso, o *software Pix4D Capture* foi capaz de criar as linhas de voo, que representam a trajetória a ser realizada pelo VANT. Junto com essas linhas foram planejados os momentos de obtenção das fotografias, que foram usadas na construção da imagem georreferenciada. A Figura 7.2 mostra as linhas de voo para construção da trajetória do VANT.

Na Figuras 7.2, os pontos em vermelho representam os locais onde foram obtidas as imagens usadas na construção da imagem georreferenciada. A trajetória realizada pela aeronave percorreu estes pontos. Junto com os pontos em vermelho também é possível ver marcações em azul, no formato de cruz. Estas marcações em azul correspondem a pontos de controle georeferenciados, que servem para auxiliar a navegação e também para identificação das coordenadas dos pixels durante o processo de construção da imagem georeferenciada.

Figura 7.2 - Linhas de voo e pontos de controle.



Fonte: Adaptado de LIMA (2018).

### 7.1.1 Estimação da trajetória por visão computacional

No primeiro experimento desta etapa, foram avaliados dois extratores de borda, também chamados de operadores. São eles o operador Sobel e o operador Canny. Além dos operadores clássicos, também utilizou-se o extrator de bordas neural. Esta abordagem consiste em treinar uma MLP para realizar o processo de extração de bordas de uma imagem binária. A arquitetura da MLP usada como extrator e bordas também foi obtida pela aplicação do MPCA. Todo o processo utilizado na extração e correlação das imagens está descrito no Capítulo 3.

Após a correlação entre a imagem georreferenciada e as imagens obtidas pelo VANT, obteve-se uma trajetória estimada pela visão computacional. Esta trajetória foi comparada com a trajetória medida pelo GPS-RTK. Cabe ressaltar que cada extrator de bordas gerou um conjunto de imagens, após a sua aplicação. Estas imagens foram correlacionadas com a imagem georreferenciada e portanto para cada operador obteve-se uma trajetória estimada.

Os resultados da estimação da posição do VANT por visão computacional mostraram que existe um grande erro associado à estimativa feita apenas por correlação cruzada, entre a imagem georreferenciada e as imagens capturadas pelo VANT.

O pior resultado foi obtido após a extração de bordas utilizando a MLP, neste caso o erro médio na estimativa da trajetória foi de 3.913,93cm. Para as coordenadas, o erro médio foi de 2.736,12cm para latitude e 2.511,81cm para longitude. O melhor resultado para o erro médio na estimativa da trajetória foi obtido pela extração de bordas pelo operador Canny, onde o erro médio foi de 3.311,74cm. Para o operador

Sobel o erro médio na estimativa da trajetória foi de 3.508,67cm. Claramente, para todas as estimativas de trajetória o erro médio foi alto. A Tabela 7.1 mostra as incertezas obtidas para cada operador aplicado na extração de bordas. As duas primeiras linhas, na Tabela 7.1, mostram o erro médio para ambas as coordenadas e a última linha o erro médio para estimação da trajetória. As incertezas foram calculadas comparando as medidas obtidas a partir do GPS-RTK.

Tabela 7.1 - Incertezas para navegação por visão computacional.

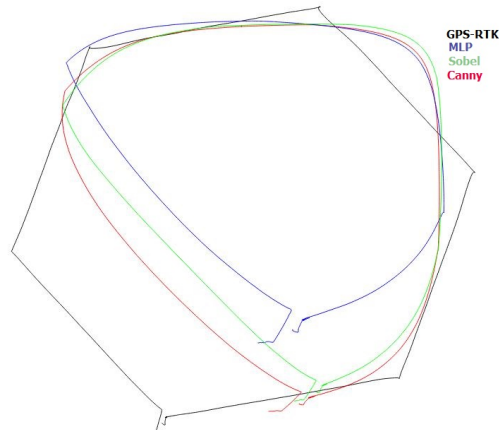
Coordenada	MLP	Sobel	Canny
Erro médio			
Latitude	2.736,12cm	1.479,65cm	1.338,95cm
Longitude	2.511,81cm	2.786,39cm	2.604,10cm
Erro médio na trajetória			
-	3.913,93cm	3.508,67cm	3.311,74cm

Uma vez que se tenha os resultados das estimações por visão computacional torna-se interessante visualizar a trajetória estimada. Isto é apresentado na Figura 7.3. Na Figura 7.3, a linha preta representa a trajetória medida pelo GPS-RTK, a linha azul mostra a trajetória estimada após a extração de bordas pela MLP, a linha verde mostra a trajetória estimada após a extração de bordas pelo operador Sobel e a linha vermelha mostra a trajetória estimada após a extração de bordas pelo operador Canny. Pela Figura 7.3 percebe-se como foi a proximidade obtida pela visão computacional para cada extrator de bordas e evidencia-se que o operador Canny obteve o melhor resultado.

Na literatura, existem três métricas disponíveis para avaliação dos resultados da estimativa por visão computacional (BRAGA, 2018). As métricas são *Error Good Matching* (EGM), *Standard Deviation Good Matching* (SDGM) e *Good Matching* (GM) (CONTE; DOHERTY, 2008; BRAGA et al., 2016; BRAGA, 2018). Estas métricas foram aplicadas nos estudos do grupo de pesquisa do WITAS e em outros trabalhos. Contudo aqui elas não são usadas, pois foram desenvolvidas com base no erro médio do GNSS embarcado no VANT usado na Suécia, que possui sensores com precisão muito diferente comparados aos sensores do VANT usado no Brasil (PAULINO, 2019). Assim manteve-se o erro médio da estimação, comparado ao GPS-RTK, como métrica para avaliação do resultado da estimação por visão computacional.

Dado que o resultado da estimação por visão computacional obteve um erro alto, um segundo experimento foi realizado no intuito de melhorar o resultado da estimação

Figura 7.3 - Trajetória estimada por visão computacional.



Fonte: Produção do autor.

por visão computacional. Na literatura, uma das formas aplicadas para melhorar o resultado da estimação da posição, por visão computacional, é aplicar filtros, antes do processo de extração de bordas (BRAGA, 2018). O objetivo de aplicar filtros é melhorar a qualidade da imagem digital, diminuindo a presença de ruídos causados na aquisição das imagens pelo sensor imageador (PEDRINI; SCHWARTZ, 2007). No Capítulo 3 é informado que a filtragem pode ser aplicada no domínio da frequência ou no domínio espacial.

Uma vez que existe a possibilidade de escolher entre estes dois domínios buscou-se então as soluções mais adequadas para o problema de correlação entre duas imagens (BRAGA, 2018), para navegação de VANT. O tipo de filtragem que se destaca em trabalhos sobre navegação autônoma de VANT é sobre o domínio espacial, inclusive este tipo de filtragem foi aplicado nos trabalhos do grupo de pesquisa do WITAS e no grupo de pesquisa do IEAv (CONTE; DOHERTY, 2008; SILVA, 2016; BRAGA, 2018).

Um dos filtros que se destaca na melhora de imagens digitais é o filtro Gaussiano (GEDRAITE; HADAD, 2011; JAIN; GUPTA, 2015; JULIO et al., 2015; WANG; HSIEH, 2018; BRAGA, 2018). Este filtro possui uma boa capacidade de suavização e remoção de ruídos em imagens digitais e por isso é um dos mais usados (WANG; HSIEH, 2018; BRAGA, 2018). Além disso, na literatura encontram-se experimentos que utilizaram o filtro Gaussiano para melhorar o processo de odometria visual (BRAGA et al., 2016; BRAGA, 2018), a odometria visual é uma das formas usada para localização

de VANT baseada em processamento de imagens (ROOS, 2018; REBOUÇAS, 2019; LEE et al., 2019). Por isso, aplicou-se o filtro Gaussiano sobre as imagens capturadas pela aeronave, com o objetivo de verificar se há melhora na estimação da posição por visão computacional após a aplicação de uma técnica de filtragem.

O resultado após a aplicação do filtro Gaussiano melhorou a estimação da trajetória por visão computacional para a maioria dos extratores de bordas usados anteriormente. Contudo, para o operador Canny o erro médio na estimação da trajetória foi piorado, anteriormente o erro médio obtido foi de 3.311,74cm, neste experimento o erro médio passou a ser 3.617,96cm, o que representa uma piora de 9% na estimação da trajetória. A maior melhora ocorreu para o operador Sobel, antes seu erro médio na estimação da trajetória era de 3.508,67cm, após a filtragem este erro passou a ser 3.078,74cm, o que representa uma melhora de 12%. As incertezas para este experimento são mostradas na Tabela 7.2.

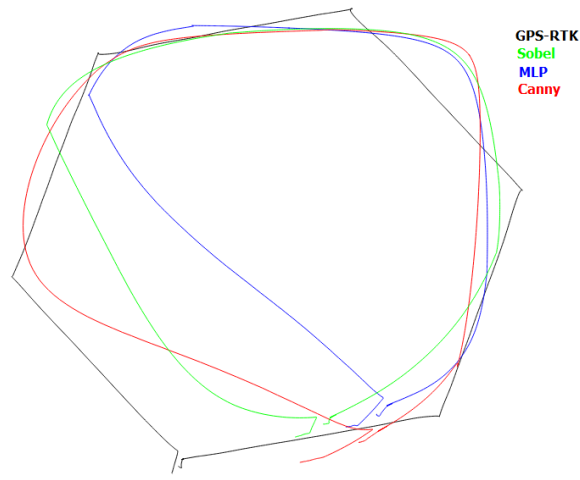
A Tabela 7.2, além de mostrar as incertezas do experimento atual, também mostra a comparação com o experimento antes de aplicar filtragem nas imagens. O experimento antes de se aplicar filtragem, foi o nono experimento realizado, por isso na Tabela 7.2 seus valores estão na coluna denominada como **EX9**. Já o experimento com aplicação de filtragem é o décimo ocorrido, portanto seus valores estão na coluna denominada de **EX10**, seguindo o padrão de nomenclatura utilizado desde o Capítulo 6. Na coluna denominada de diferença é apresentado a porcentagem de aumento ou redução para a incerteza, sendo que valores positivos representam aumento e valores negativos representam redução.

Uma vez que se tenha as incertezas para a estimação da trajetória por visão computacional, após a aplicação do filtro Gaussiano, torna-se interessante visualizar a trajetória estimada. Isto é apresentado na Figura 7.4. Na Figura 7.4, a linha preta representa a trajetória medida pelo GPS-RTK, a linha azul mostra a trajetória estimada após a extração de bordas pela MLP, a linha verde mostra a trajetória estimada após a extração de bordas pelo operador Sobel e a linha vermelha mostra a trajetória estimada após a extração de bordas pelo operador Canny, ressalta-se que a extração de bordas ocorreu após a aplicação do filtro Gaussiano. Pela Figura 7.4 percebe-se como foi a proximidade obtida pela visão computacional para cada extrator de bordas, após aplicação de filtragem.

Tabela 7.2 - Incertezas para navegação por visão computacional, após filtragem.

Coordenada	EX9	EX10	Diferença
MLP - Erro médio			
Latitude	2.736,12cm	1.656,67cm	-39,45%
Longitude	2.511,81cm	3.154,50cm	20,37%
Erro médio na trajetória			
-	3.913,93cm	3.818,18cm	-2,44%
Sobel - Erro médio			
Latitude	1.479,65cm	1.379,50cm	-6,76%
Longitude	2.786,39cm	2.483,74cm	-10,86%
Erro médio na trajetória			
-	3.508,67cm	3.078,74cm	-12,25%
Canny - Erro médio			
Latitude	1.338,95cm	1.227,97cm	-8,28%
Longitude	2.604,10cm	3.067,18cm	17,78%
Erro médio na trajetória			
-	3.311,74cm	3.617,96cm	9,24%

Figura 7.4 - Trajetória estimada por visão computacional, após filtragem.



Fonte: Produção do autor.

### 7.1.2 Considerações sobre a estimativa da trajetória por visão computacional

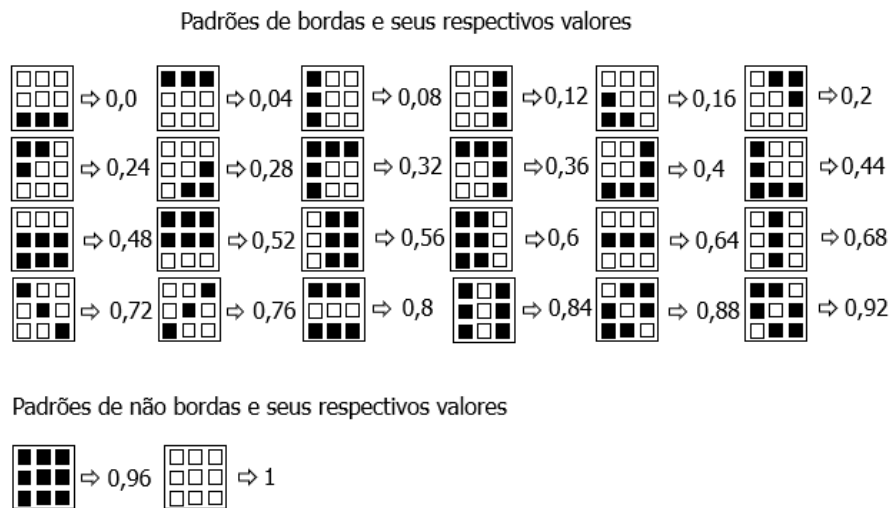
Os experimentos realizados com visão computacional objetivaram verificar se existe e qual é o valor do erro médio associado a estimativa da trajetória do VANT, por processamento de imagens. Este erro médio ficou evidenciado nos dois experimentos

realizados. Contudo, cabe ressaltar que o erro médio para estimação da trajetória por visão computacional recebe influência de fatores como o tipo do sensor imageador, posicionamento da câmera embarcada, tipo da câmera usada, quantidade de imagens utilizadas para correlação cruzada, intensidade luminosa entre outros (AL-KAFF et al., 2018). Por isso, não é possível afirmar que as incertezas obtidas nos experimentos anteriores serão sempre as mesmas encontradas em novos experimentos, dado que podem ser influenciadas pelos equipamentos utilizados ou por condições naturais.

Anteriormente, mencionou-se que há mais de um tipo de técnica que pode ser aplicada para a estimação da trajetória de uma aeronave, por processamento de imagem. Então cabe dizer que correlação cruzada entre duas imagens foi escolhida porque é uma técnica já aplicada em outros trabalhos e que apresentou bons resultados (SILVA et al., 2015; SILVA, 2016; BRAGA et al., 2016; BRAGA, 2018).

Com relação ao extrator de bordas neural, torna-se importante informar quais foram os padrões de bordas utilizados para o treinamento da MLP. Nesta etapa utilizou-se 24 padrões de bordas e 2 padrões para ausência de bordas. A Figura 7.5 mostra os padrões de bordas utilizados e seus valores correspondentes.

Figura 7.5 - Padrões de bordas utilizados para treinamento do extrator de bordas neural.



Fonte: Produção do autor.

Há outras formas de padrões de bordas para treinamento de uma MLP, para extração de bordas (SILVA, 2016; BRAGA, 2018). Geralmente são formados padrões que seguem



uma combinação de 3 *pixels* representado borda e os demais representando ausência de bordas, dentro de uma matriz 3x3 *pixels*. Os padrões de bordas apresentados na Figura 7.5 foram descritos e utilizados em outros trabalhos, além disso, apresentaram resultados importantes e por isso foram escolhidos (BRAGA, 2018).

Apesar das incertezas encontradas nos experimentos serem influenciadas pelas condições de voo e equipamentos embarcados, é possível concluir que estimar o posicionamento do VANT apenas por visão computacional é perigoso, pois as incertezas apresentadas nas Tabelas 7.1 e 7.2 mostram que há um erro médio grande para estimação da trajetória.

Torna-se importante evidenciar que o cálculo da correlação entre as imagens não foi realizado sobre toda a extensão da imagem georreferenciada, que está mostrada na Figura 7.1. A Figura 7.1 foi recortada e o recorte feito foi no formato quadrado centrado na imagem hexagonal que representa a área do voo realizado pelo VANT. As dimensões tomadas para o recorte foram de 300m por 300m. Desta forma delimitou-se o espaço na imagem georreferenciada utilizada para correlação cruzada.

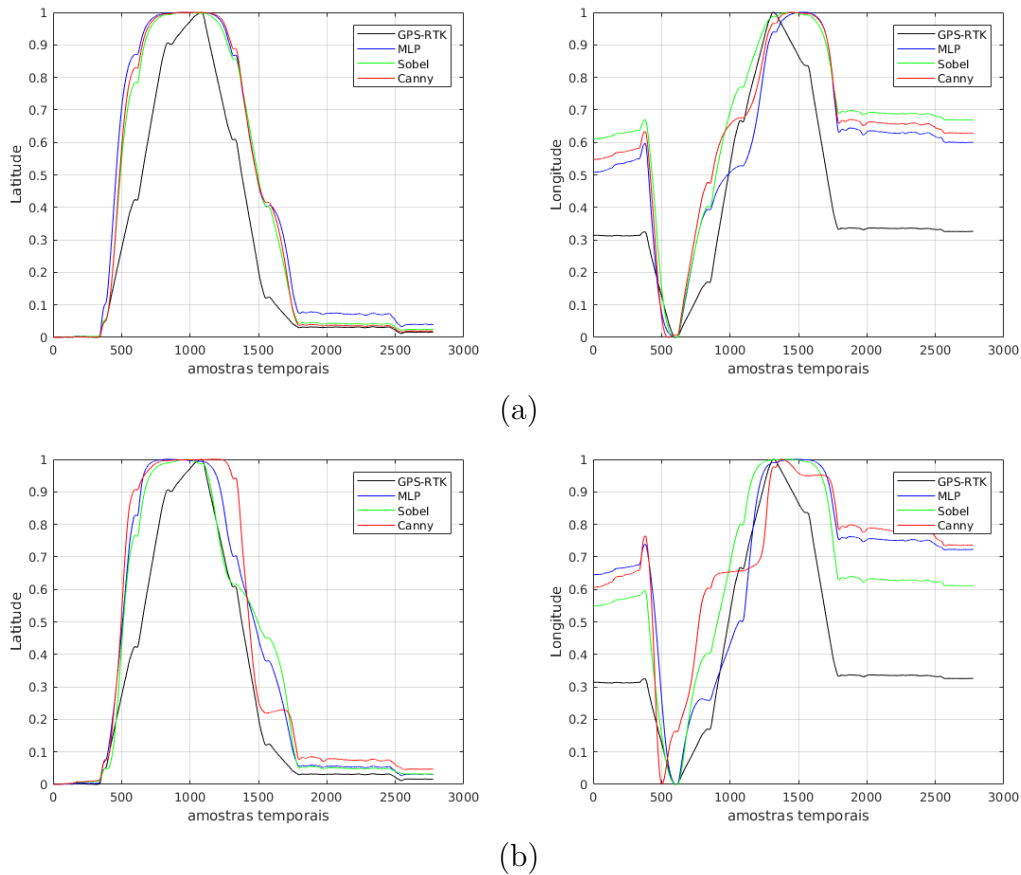
Dado que é apresentado, nas Figuras 7.3 e 7.4, a trajetória estimada pela visão computacional e também a trajetória medida pelo GPS-RTK, torna-se interessante visualizar a curva das estimações para cada coordenada. Isto é apresentado na Figura 7.6. Na Figura 7.6 estão quatro gráficos, os dois primeiros mostram a estimação para ambas as coordenadas, no experimento onde o filtro gaussiano não foi aplicado, os demais mostram a estimação para ambas as coordenadas no experimento onde aplicou-se o filtro Gaussiano. Para facilitar a comparação entre a Figura 7.6 com as demais Figuras 7.3 e 7.4, adotou-se o mesmo esquema de cores na Figura 7.5. Cabe destacar que na Figura 7.6 as informações estão normalizadas, isto é devido a uma solicitação de sigilo de informações do IEAv, contudo a normalização não altera o formato original das curvas de estimações para cada coordenada.

Além das curvas, que mostram a estimação para cada coordenada também torna-se interessante mostrar algumas das imagens usadas para fazer a correlação com a imagem georreferenciada. Estas imagens são mostradas na Figura 7.7. As imagens com bordas extraídas, na Figura 7.7, correspondem ao resultado da extração de bordas feita pela MLP, após aplicação do filtro Gaussiano.

Escolheu-se mostrar algumas das imagens com bordas extraídas pela MLP porque o resultado da estimação da trajetória obtido pela correlação cruzada dessas imagens com a imagem georreferenciada será usado na segunda etapa de experimentos com



Figura 7.6 - Estimação de cada coordenada por visão computacional.



- (a) Resultado da estimação para ambas as coordenadas, sem aplicação do filtro Gaussiano.  
(b) Resultado da estimação para ambas as coordenadas, com aplicação do filtro Gaussiano.

Fonte: Produção do autor.

a inclusão das informações dos sensores inerciais. As outras imagens não foram mostradas porque mostrar todas as imagens não seria visivelmente confortável ao leitor.

Apesar do resultado da estimação da trajetória por visão computacional, quando usadas as imagens geradas após a extração de bordas pela MLP, não ter sido o melhor resultado, este extrator de bordas foi escolhido para a próxima etapa, porque na literatura encontram-se resultados importantes que mostram a aplicabilidade de redes neurais como extrator de bordas em problemas de navegação autônoma de VANT (SILVA et al., 2015; SILVA, 2016; BRAGA, 2018). Além disso, este extrator também foi implementado em *hardware* dedicado com FPGA.

Figura 7.7 - Imagens usadas para estimação da trajetória, utilizando a correlação cruzada.



(a) As imagens estão em pares, à esquerda está a imagem capturada pela câmera embarcada e à direita a respectiva imagem com bordas extraídas.

Fonte: Produção do autor.

## 7.2 Inclusão dos dados inerciais

Pelos resultados obtidos anteriormente, a estimação da trajetória do VANT por visão computacional por si só não é suficiente. Na intenção de melhorar esta estimação e seguindo a metodologia proposta, torna-se necessário realizar a fusão de dados com outros sensores embarcados na aeronave.

Nos experimentos realizados no Capítulo 6 foram fusionados 12 tipos de dados, obtidos por sensores embarcados. Os tipos de dados usados foram latitude e longitude obtidos pelo sensor inercial, 3 taxas angulares, 3 medidas de acelerômetro, medidas de velocidade norte, oeste, subida e por último altura. A aeronave utilizada no Brasil também possui sensores que captam estes tipos de dados (DJI, 2020a). Assim manteve-se estes 12 tipos aplicados na fusão de dados, nos experimentos no Brasil.

Nos experimentos no Brasil a fusão de dados foi realizada por uma MLP autoconfigurada, semelhante ao realizado nos experimentos do Capítulo 6. Apesar dos experimentos de visão computacional mostrarem que os operadores Canny e Sobel obtiveram melhores resultados que a MLP, utilizou-se na fusão de dados a estimação da trajetória feita pela MLP, obtida no último experimento com visão computacional - as estimações da trajetória obtidas com os demais operadores não foram utilizadas na fusão de dados.

Cabe destacar que no Capítulo 6 foi realizado um estudo sobre o impacto da organização dos dados no treinamento da MLP. Foram avaliadas formas de selecionar os dados do conjunto de treinamento, com a intenção de melhorar o erro médio da estimação da trajetória. Este estudo, mostrou que usar uma porcentagem igual ou menor que 30% do total de dados, para treinamento da MLP, não produz uma estimação da trajetória adequada. Também foi apresentado que usar a abordagem de validação cruzada *k-fold*, pode ajudar a selecionar melhor os dados de treinamento, e com isso aprimorar a estimação da trajetória. Desta forma, nos experimentos realizados no Brasil, fez-se a seleção dos dados para treinamento da MLP de modo igual ao estudo do Capítulo 6.

Também cabe destacar que a os dados obtidos durante o voo na Suécia, foram coletados em tempos desunidos. Por isto no Capítulo 6, antes dos estudos com fusão de dados, foi necessário selecionar os dados relacionados temporalmente, o que provocou uma redução na quantidade total de dados disponível para treinamento. No caso do voo no Brasil, a fabricante do VANT fornece uma ferramenta, um software, que realiza essa seleção de maneira automática. Desta forma, nenhuma pré-seleção ou tratamento afim de interpolar as informações coletadas pelos sensores inerciais foi necessário.

### 7.2.1 Fusão de dados

Os dados obtidos da estimação por visão computacional foram fusionados com os dados obtidos pelos sensores embarcados. Esta fusão de dados melhorou a estimação

da trajetória do VANT de maneira significativa. As incertezas obtidas para a fusão de dados são mostradas na Tabela 7.3.

Tabela 7.3 - Incertezas para a fusão de dados.

Coordenada	Visão computacional	Fusão de dados	Redução
Erro médio			
Latitude	2.736,12cm	475,11cm	82,63%
Longitude	2.511,81cm	270,45cm	89,23%
Desvio			
Latitude	1.245,94cm	37,22cm	97,01%
Longitude	1.084,24cm	121,91cm	88,75%
Variância			
Latitude	1.552.376,37cm	1.385,42cm	99,91%
Longitude	1.204.122,29cm	14.862,29cm	98,76%
Erro médio na trajetória			
-	3.913,93cm	555,37cm	85,81%

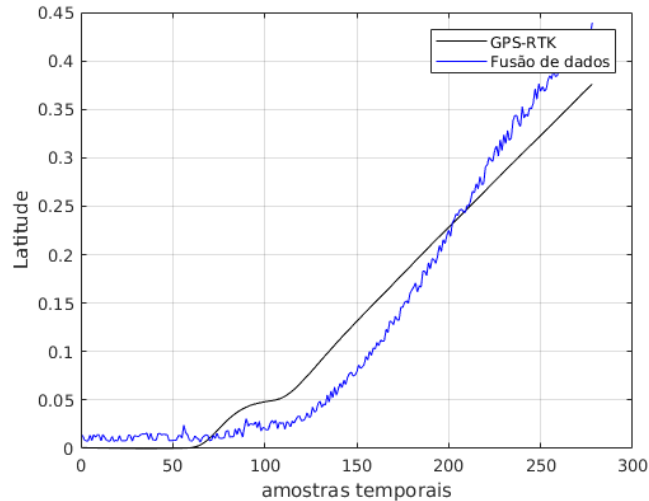
Além de mostrar as incertezas para o experimento com fusão de dados, a Tabela 7.3 mostra a comparação com o resultado da estimação da trajetória por visão computacional, ressaltando que a comparação é feita com as incertezas da estimação por visão computacional obtidas pelo extrator de bordas MLP. Junto com erro médio, também foram comparadas as incertezas que denotam intervalo de confiança.

Para a estimação de cada coordenada houve uma redução do erro médio de mais de 80%. Com a fusão de dados, o erro médio de estimação para coordenada latitude foi de 475cm e para coordenada de longitude foi de 270cm, valores significativamente menores do que os encontrados apenas por visão computacional. Com relação as incertezas que denotam intervalo de confiança as reduções também foram expressivas, ultrapassando 90% de redução, o que indica aumento da confiabilidade nas estimações. As melhoras obtidas tanto nas estimações de cada coordenada como também no intervalo de confiança influenciaram positivamente a estimação na trajetória, que para fusão de dados obteve o erro médio de 555cm, 85% menor que o erro médio de estimação da trajetória considerando apenas a visão computacional.

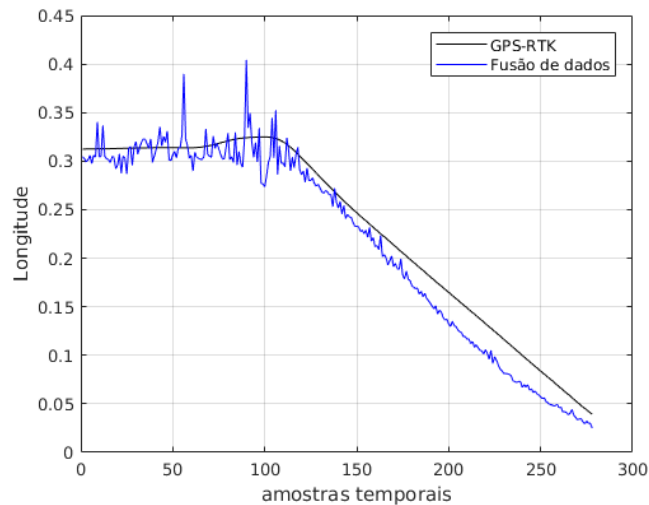
As incertezas para a fusão de dados, mostradas na Tabela 7.3, foram calculadas sobre o conjunto de generalização, ou seja, sobre o conjunto usado para testar o fusor neural. Assim torna-se interessante visualizar a estimação feita pela MLP para este conjunto de dados, isto é apresentado na Figura 7.8. Na Figura 7.8 são mostradas as estimações para cada coordenada. Cabe ressaltar que este conjunto de dados

equilave a 10% do total de dados e foi obtido com a mesma técnica aplicada no Capítulo 6.

Figura 7.8 - Estimações obtidas pela fusão de dados com MLP autoconfigurada.



(a)



(b)

(a) Valores para a coordenada de latitude. (b) Valores para a coordenada de longitude.

Fonte: Produção do autor.

Percebe-se na Figura 7.8, que o resultado da estimação foi afetado pela presença de ruído, os valores das estimações para cada coordenada mostram variações que não foram sentidas pelo GPS-RTK. Isto é um indicativo que é possível melhorar o

resultado da estimação do fusor neural com aplicação de filtragem sobre os dados inerciais.

No Capítulo 6 evidenciou-se o impacto negativo que a presença de ruído pode provocar nas estimações do fusor neural, também foram demonstradas duas formas de reduzir este impacto. A primeira forma foi através da remoção das variações rápidas nas medidas obtidas pelo INS e a segunda forma foi através da aplicação de um filtro linear. Portanto, este mesmo processo foi aplicado aos dados obtidos no Brasil.

Torna-se importante ressaltar que nos experimentos com filtros lineares, feitos no Capítulo 6, o que obteve o melhor resultado foi o Savitzky-Golay. Por isso, o filtro Savitzky-Golay foi aplicado nesta etapa, os demais filtros não foram testados. Além disso, utilizou-se o mesmo tamanho de janela deslizante, aplicado anteriormente no Capítulo 6, desta forma manteve-se a diferença de aproximadamente 3cm entre as coordenadas originais e as coordenadas filtradas utilizados para o treinamento do fusor neural.

Uma vez que se tem os dados inerciais filtrados, estes dados foram utilizados no treinamento de uma nova MLP, esta nova MLP foi usada novamente como fusor de dados sobre o mesmo conjunto testado anteriormente. Cabe ressaltar que nenhuma outra técnica de filtragem foi aplicada sobre as imagens usadas na estimação por visão computacional. As imagens usadas na estimação por visão computacional já passaram por uma etapa de filtragem anteriormente. Portanto, nesta etapa, aplicou-se somente filtragem nos dados inerciais.

A fusão de dados obteve melhora na estimação da trajetória, após a aplicação de filtragem nos dados inerciais. O erro médio na trajetória, para fusão de dados, era de 555,37cm, após a filtragem este erro médio diminuiu em mais de 70% e o novo valor para o erro médio foi de 118cm.

A redução no erro médio de estimação da trajetória implica que houve redução no erro médio para estimação de cada coordenada. A redução no erro médio para estimação da coordenada latitude foi de 91,30%, já para a coordenada longitude a redução no erro médio foi de 58%. Houve também redução nas incertezas que denotam intervalo de confiança. As incertezas para este experimento são mostradas na Tabela 7.4.

Além de mostrar as incertezas, a Tabela 7.4 faz uma comparação entre este experimento e o experimento anterior. Como o experimento anterior foi o décimo primeiro

experimento realizado, seus valores são apresentados na coluna denominada **EX11**, na Tabela 7.4. O experimento com filtragem aplicada nos dados inerciais foi o décimo segundo realizado, assim seus valores são apresentados na coluna denominada **EX12**, seguindo o padrão de nomenclatura usado na maioria das tabelas anteriores.

Tabela 7.4 - Incertezas para a fusão de dados, após filtragem nos dados inerciais.

Coordenada	EX11	EX12	Redução
Erro médio			
Latitude	475,11cm	41,29cm	91,30%
Longitude	270,45cm	111,33cm	58,83%
Desvio			
Latitude	37,22cm	6,38cm	82,85%
Longitude	121,91cm	4,17cm	96,57%
Variância			
Latitude	1.385,42cm	40,78cm	97,05%
Longitude	14.862,29cm	17,40cm	99,88%
Erro médio na trajetória			
-	555,37cm	118,89cm	78,59%

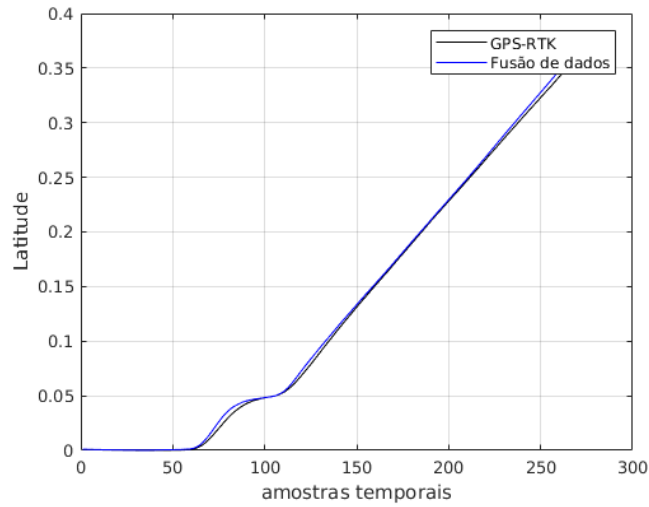
Uma vez que se tenha as incertezas para fusão de dados, após aplicação de filtragem nos dados inerciais, torna-se interessante visualizar a estimativa feita para cada coordenada, isto é apresentado na Figura 7.9. Percebe-se pela Figura 7.9 que o resultado da estimativa, para cada coordenada, aproximou-se dos valores obtidos pelo GPS-RTK. Portanto é possível comprovar visualmente porque a filtragem contribuiu para a redução do erro médio nas estimativas. Da mesma forma que na Figura 7.8, os dados apresentados na Figura 7.9 estão normalizados para atender a solicitação de sigilo do IEAv.

### 7.2.2 Considerações sobre a inclusão dos dados inerciais

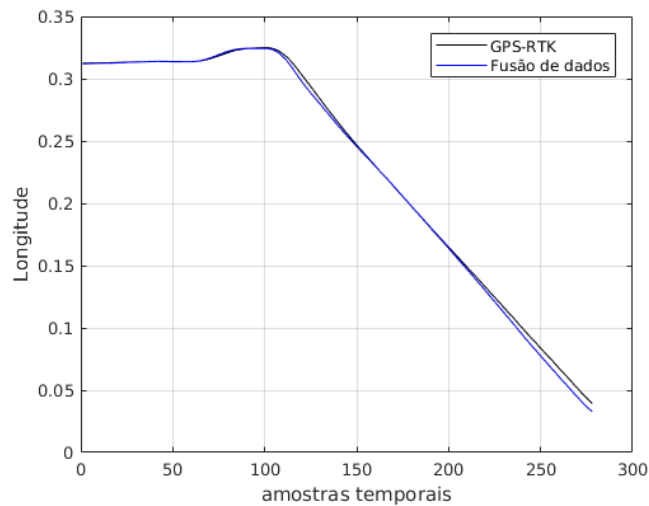
A etapa de inclusão dos dados inerciais focou-se em melhorar a estimativa da trajetória através da fusão de dados, uma vez que o resultado da estimativa da trajetória por visão computacional apresentou valores distantes dos obtidos pelo GPS-RTK. Além disso, as incertezas que denotam intervalo de confiança, para a estimativa da trajetória por visão computacional, foram altas, o que indica insegurança em estimativas feitas apenas por processamento de imagens.

A fusão de dados combinou os dados inerciais com a estimativa da posição por visão computacional e isto provocou uma melhora significativa na estimativa do posicio-

Figura 7.9 - Estimacões obtidas pela fusão de dados com MLP autoconfigurada.



(a)



(b)

(a) Valores para a coordenada de latitude. (b) Valores para a coordenada de longitude.

Fonte: Produção do autor.

namento final da aeronave. Além de melhorar a estimacão da posicão do VANT, a etapa de inclusão dos dados inerciais mostrou que uma MLP autoconfigurada consegue realizar o trabalho antes feito por um FK, uma vez que os resultados mostraram que a estimacão da posicão do VANT, feita pelo fusor neural, se aproximou das medidas de um equipamento GPS de precisão alta. Além disso, as incertezas que denotam intervalo de confiança mostraram que as estimacões da posicão da aeronave,



feitas pelo fusor neural, possuem garantia de proximidade, quando comparadas com as incertezas que denotam intervalo de confiança obtidas pela visão computacional, isto pode ser visto nas medidas de variância e desvio mostradas anteriormente na Tabela 7.4.

Com relação a não utilizar as estimações da trajetória encontradas pelos outros extratores de bordas, cabe informar que um dos objetivos do trabalho é desenvolver uma técnica, composta em sua maior parte, por processamento utilizando redes neurais. Uma vez que o resultado da fusão de dados, utilizando a estimação da trajetória pelo extrator de bordas neural, obteve um ótimo resultado, ou seja, medidas de latitude e longitude próximas as obtidas pelo GPS-RTK, decidiu-se então que não seria necessário testar outros extratores de bordas.

É importante informar que a fusão de dados apresentada neste capítulo e no capítulo anterior, foi realizada em CPU, ou seja, até esta etapa, de inclusão dos dados inerciais, não se usou uma FPGA ou algum outro *hardware* dedicado - o objetivo foi, primeiramente, validar o fusor neural. O computador utilizado, nesta etapa, foi um *Intel Core i5-6200U* com 2.30GHz, também 8GB de RAM e sistema operacional de 64bits.

Anteriormente, foi informado que a arquitetura da MLP utilizada na fusão de dados foi obtida a partir da aplicação do MPCA. Então, cabe mostrar qual foi a arquitetura aplicada ao fusor neural. Esta arquitetura é resumida na Tabela 7.5. A Tabela 7.5 mostra informações que descrevem a arquitetura da rede neural utilizada na fusão de dados e também traz informações dos parâmetros de treinamento, que foram utilizados pelo MPCA - para a fusão de dados, o MPCA obteve uma arquitetura de MLP com duas camadas escondidas. Cabe repetir que o objetivo na aplicação do MPCA é encontrar uma arquitetura ótima, no sentido de se ter o menor número de neurônios e camadas escondidas que garantem um resultado otimizado na aplicação da rede neural.

Por fim, a vantagem em usar uma MLP no lugar do FK está em que uma MLP possui complexidade computacional menor que um FK. A menor complexidade da MLP significa menor custo computacional, ou seja, menor exigência e gasto de energia de um processador na aplicação da MLP como fusor de dados. Além disso, como já informado anteriormente, a MLP não dependeu da dinâmica da aeronave. Não depender do modelo de VANT significa que a metodologia pode ser aplicada com mais liberdade sem depender de componentes específicos, fabricantes específicos e com isso pode ser aplicada a aeronaves com custo de fabricação ou montagem menor.

Tabela 7.5 - Parâmetros selecionados pelo MPCA para o primeiro experimento.

Parâmetro	Valor ou Tipo
Épocas de treinamento	300
Número de camadas escondidas	2
Neurônios na primeira camada escondida	12
Neurônios na segunda camada escondida	10
Neurônios na camada de saída	2
Taxa de Aprendizagem	0,7
Taxa de momento	0,5
Função de ativação	sigmoid

### 7.3 Desenvolvimento de um coprocessador em *hardware* dedicado

Anteriormente, foi informado que existem vários tipos de VANTs (HUTTUNEN, 2017; BRAGA, 2018; PAULINO, 2019). Inclusive, que há mais de uma definição para esta categoria de aeronave (BRAGA, 2018; PAULINO, 2019; LIMA, 2018; AGÊNCIA NACIONAL DE AVIAÇÃO CIVIL, 2020). Apesar das definições é possível encontrar características que são comuns a maioria dos tipos de VANTs. Dentre essas características pode-se destacar que é comum para VANT usar uma ou mais baterias como principal ou até única fonte de energia, que mantém o funcionamento da aeronave durante o voo (VALAVANIS; VACHTSEVANOS, 2014).

Essas baterias são responsáveis por fornecer energia para cada componente embarcado na aeronave, portanto, o tempo de duração da energia armazenada nas baterias do VANT está diretamente relacionado ao consumo de energia de cada componente. Isto significa que componentes de baixo desempenho, ou seja, componentes que consomem muita energia para funcionar diminuem o tempo de autonomia em voo da aeronave.

Outro fator a ser considerado, sobre o quanto dura a energia das baterias de um VANT, é a capacidade do VANT de carregar peso, sua capacidade de carga, ou seja, sua capacidade de transportar algum objeto. Ainda que os componentes utilizados na fabricação do VANT sejam de alto desempenho, colocar algo para que seja transportado pela aeronave irá, inevitavelmente, aumentar o consumo de energia em seus motores e isto também pode diminuir o tempo de autonomia em voo. Portanto, é fácil entender que o transporte de carga juntamente com o peso da própria aeronave afeta o tempo de voo, devido ao consumo de energia.

Nos estudos realizados pelo grupo de pesquisa do WITAS, foi embarcado no VANT

três computadores, ou seja, três CPUs (CONTE; DOHERTY, 2009). Então estima-se que o consumo de energia para garantir a autonomia do VANT utilizado na Suécia foi elevado, uma vez que foi necessário o funcionamento de três CPUs para realizar todo o processo de aquisição, tratamento e fusão de dados utilizando o FK.

Um dos objetivos deste trabalho é fornecer uma técnica para ser embarcada, de modo a substituir o FK e garantir o resultado satisfatório para estimação da posição do VANT, utilizando a fusão de dados. Neste sentido as sessões anteriores mostraram que é possível substituir o FK por uma MLP autoconfigurada - diminuindo a complexidade computacional. Mas, substituir o FK por uma MLP vai além de diminuir a complexidade computacional. É possível a implementação da MLP autoconfigurada, para embarque, em dispositivos menores que CPUs e conseqüentemente com consumo de energia menor, uma vez que estes dispositivos são mais leves e exigem menos energia para seu funcionamento do que CPUs, como as que foram utilizadas no experimento na Suécia (BRAGA, 2018; VITOR et al., 2020).

Com relação ao embarque de uma MLP num VANT, para fusão de dados, cabe ressaltar que de nada adianta reduzir o peso do dispositivo embarcado ou o consumo de energia se a capacidade de processamento não for igual ou superior ao processamento de uma CPU, sob pena de não ser possível realizar a fusão de dados em tempo hábil, o que resultaria numa estimação da posição atrasada e no descontrole da aeronave em seguir a trajetória desejada. Neste sentido buscou-se alternativas na literatura que permitissem atender este requisito de desempenho. A alternativa que melhor se adequa é utilizar *Field Programmable Gate Array* (FPGA) (VITOR et al., 2020).

O objetivo de desenvolver a MLP autoconfigurada em FPGA é fornecer um dispositivo leve e com velocidade de processamento alta, para embarque no VANT (VITOR et al., 2020). Este dispositivo atua como um coprocessador, acoplado a malha de controle do VANT ou ainda como um periférico com interação direta com a controladora de voo. Além da fusão de dados, é possível atribuir ao coprocessador a tarefa de extração de bordas - tarefa também feita por uma MLP autoconfigurada como demonstrado nas sessões anteriores.

### 7.3.1 Fusão de dados no computador *Raspberry pi*

Para o primeiro desenvolvimento do fusor neural em FPGA, foi utilizado um computador *single-board*, pequeno e de baixo custo. Este computador é denominado *Raspberry Pi* (RASPBERRY PI FOUNDATION, 2020), especificamente utilizou-se o modelo

### *Raspberry Pi-B1.*

Este dispositivo foi escolhido porque há trabalhos na literatura que utilizaram modelos do computador *Raspberry Pi*, juntamente com VANT, em diversas aplicações, e como componente embarcado para extração de bordas e processamento de imagens, estes trabalhos demonstraram que este computador é capacitado para integração com VANTs (DARYANAVARD; HARIFI, 2018; BRAGA, 2018; YAMAZAKI et al., 2019; GUAN et al., 2020).

As principais características do computador *Raspberry Pi-B1* são (BRAGA, 2018):

- Peso de 45g;
- Comprimento de 8,5cm e largura de 5,6cm;
- Consumo energético de 3,5wats e 700mA;
- Processador ARM1176JZF-S, single core com 700MHz e arquitetura de 32bits;
- Memória de 512MB;
- Duas portas *Universal Serial Bus* (USB) 2.0;
- Uma saída de vídeo *High-Definition Multimedia Interface* (HDMI);
- Uma *Graphics Processing Unit* (GPU) com 250MHz;
- Um encaixe para cartão micro-sd.

O computador *Raspberry Pi* utiliza um cartão de memória como disco rígido e neste dispositivo, é possível instalar sistemas operacionais baseados em distribuições *Linux*, como o *Debian*, ou ainda uma versão específica o sistema operacional *Windows* - estes sistemas são fornecidos pelo fabricante do computador e desenvolvidos especificamente para controle do dispositivo (RASPBERRY PI FOUNDATION, 2020).

Em sua estrutura, o computador *Raspberry Pi* possui um conjunto de 26 pinos. Estes pinos são ligados a um barramento utilizado para comunicação do computador *Raspberry Pi* com periféricos. Especificamente, dos 26 pinos, somente 16 estão ligados a um barramento de comunicação para transferência de dados. Os demais pinos restantes são divididos em: um para *flag* de sinalização de leitura ou escrita de dados no periférico conectado ao computador, quatro são usados como fonte de

alimentação para o periférico e os restantes são conectados a referência 0v - ou seja, o terra da placa.

O fabricante do computador *Raspberry Pi* disponibiliza vários periféricos que podem ser encaixados sobre os 26 pinos disponíveis no dispositivo. Estes periféricos podem ser sensores de pressão, sensores de ultrassom, sensores de calor e até mesmo sensores que auxiliam na navegação de VANT como um GNSS (RASPERRY PI FOUNDATION, 2020). Além desses periféricos, um periférico que pode ser conectado, ao computador *Raspberry Pi* pelos aos 26 pinos, é a placa *LOGI Pi* (VALENTFX, 2020).

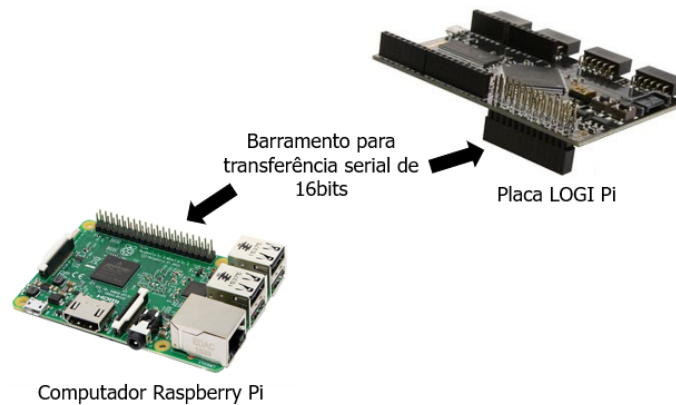
A placa *LOGI Pi* é uma plataforma de desenvolvimento FPGA que foi otimizada para uso com o computador *Raspberry Pi* (VALENTFX, 2020). Entre seus componentes, a placa *LOGI Pi* possui uma FPGA da família *Spartan 6 LX9* (VALENTFX, 2020). A combinação *LOGI Pi* e *Raspberry Pi* foi projetada com os principais atributos de design, incluindo facilidade de uso, capacidade máxima de transferência de dados e baixo custo em relação ao consumo de energia. As principais características da FPGA presente na placa *LOGI Pi* são (BRAGA, 2018):

- 9.152 elementos lógicos;
- 715 *Configurable Logic Blocks*(CLB);
- 72KB de *Random Access Memory*(RAM).

A comunicação entre o computador *Raspberry Pi* e o placa *LOGI Pi* acontece por um barramento específico denominado *Serial Peripheral Interface* (SPI). Por meio de um protocolo de transferência, desenvolvido pelo fabricante do computador *Raspberry Pi*, o SPI interconecta o barramento onde estão conectados os 16 pinos, usados para transferência de dados com a CPU do computador *Raspberry Pi*, com um componente de controle da placa *LOGI Pi*. Este componente de controle na placa *LOGI Pi* é denominado de *wishbone*.

O componente *wishbone* é um barramento destinado a comunicação da placa *LOGI Pi* com algum outro dispositivo, como o computador *Raspberry Pi*, e dado que a transferência de dados entre o computador *Raspberry Pi* e a placa *LOGI Pi* é feita por 16 pinos, então o tamanho da palavra, ou seja, o tamanho do dado, que pode ser transferido entre os dispositivos é de 16bits. A Figura 7.10 mostra os dois dispositivos usados para implementação e teste do fusor neural em FPGA, cabe destacar que a transferência de dados entre os dispositivos é feita de maneira serial.

Figura 7.10 - Primeiros dispositivos usados para implementação e teste do fusor neural em FPGA.



Computador *Raspberry Pi*-B1 e placa *LOGI Pi*.

Fonte: Adaptado de [valentfx \(2020\)](#).

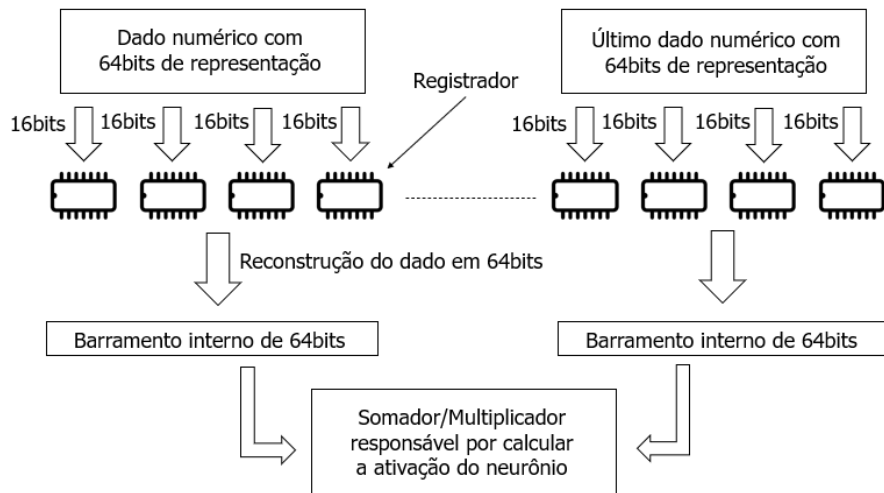
Na literatura, existem abordagens para a desenvolvimento de redes neurais em FPGA ([VITOR et al., 2020](#)). Dentre estas abordagens, destaca-se a construção de neurônios artificiais como uma composição de blocos somadores e acumuladores ([SAMBATTI et al., 2016](#); [BRAGA, 2018](#)). Contudo, esta abordagem não foi aplicada na construção do fusor neural, sendo utilizada uma abordagem que seguiu um arranjo de blocos registradores. De modo simplificado, pode-se entender um registrador como uma composição de um ou mais *flip-flops* usados para armazenar dados ([PEDRONI, 2008](#); [PIMENTA, 2017](#)). Em outras palavras, um registrador é um circuito digital como um *flip-flop* ou uma matriz de *flip-flops* que armazena um ou mais *bits* de informação digital.

Para construção de cada neurônio artificial do fusor neural em FPGA, na placa *LOGI Pi*, foram necessários 4 blocos registradores para receber e armazenar cada entrada apresentada ao neurônio e mais 4 blocos registradores para armazenar a saída da ativação do neurônio artificial. Isto foi necessário porque cada bloco registrador, na FPGA da placa *LOGI PI*, pode armazenar apenas 16bits e os valores, obtidos pelos sensores embarcados no VANT, são representados em 64bits.

Para o envio e armazenamento de cada dado para a FPGA foi necessário aplicar algumas operações sobre o dado. A primeira operação consistiu em normalizar todos os dados no intervalo  $[0, 1]$ , esta normalização é necessária para a ativação da MLP uma vez que a função de ativação corresponde a uma sigmóide. A segunda operação

foi dividir cada dado normalizado em quatro partições binárias, com 16 bits para cada partição. Assim foi possível enviar os dados para dentro do coprocessador. Uma vez dentro do coprocessador, esses dados foram remontados com a concatenação das partições binárias. A Figura 7.11 ilustra o processo necessário para envio e armazenamento de dados na FPGA.

Figura 7.11 - Processo de envio e armazenamento de dados para dentro do fusor neural.



Fonte: Produção do autor.

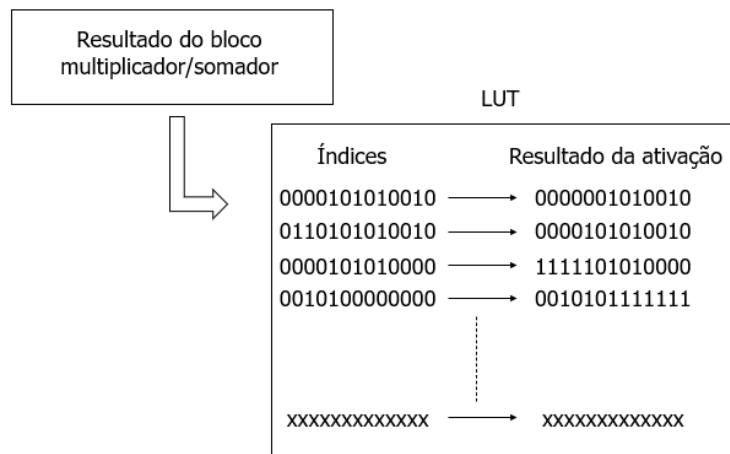
A mesma representação em 64 *bits* foi aplicada aos valores de pesos e *bias*. Contudo, uma vez que os valores de pesos e *bias* não mudam, estes valores são armazenados dentro da FPGA como constantes. Além disso, valores iguais ou semelhantes são compartilhados, ou seja, se um mesmo valor de peso é usado para mais de um neurônio, utiliza-se a mesma constante mais de uma vez.

Com relação a função de ativação do neurônio artificial, utilizou-se uma *Look up Table* (LUT) para seu desenvolvimento em FPGA. De maneira simplificada, uma LUT é uma matriz que substitui a computação em tempo de execução por uma operação de indexação matricial mais simples na memória da FPGA (PEDRONI, 2008; PIMENTA, 2017; BRAGA, 2018). Desta forma, tem-se economia no tempo de processamento, porque recuperar um valor da memória é geralmente mais rápido de que realizar um cálculo (BRAGA, 2018).

As LUTs podem ser pré-calculadas e armazenadas como constantes num registrador

ou bloco lógico ou ainda inicializadas, ou seja, pré-buscadas como parte da fase de inicialização da configuração da FPGA. Na implementação do fusor neural em FPGA foi gerada uma LUT armazenada como constantes num bloco lógico. Cada constante representa um índice na LUT, que é associado a um valor binário, este valor corresponde ao resultado da ativação do neurônio. A Figura 7.12 ilustra o esquema de como usar uma LUT para implementar a função de ativação do neurônio artificial. Na Figura 7.12 os valores em binários apresentados não representam os valores reais utilizados no experimento, são apenas valores ilustrativos.

Figura 7.12 - Implementação da função de ativação do neurônio com uma LUT.



Fonte: Produção do autor.

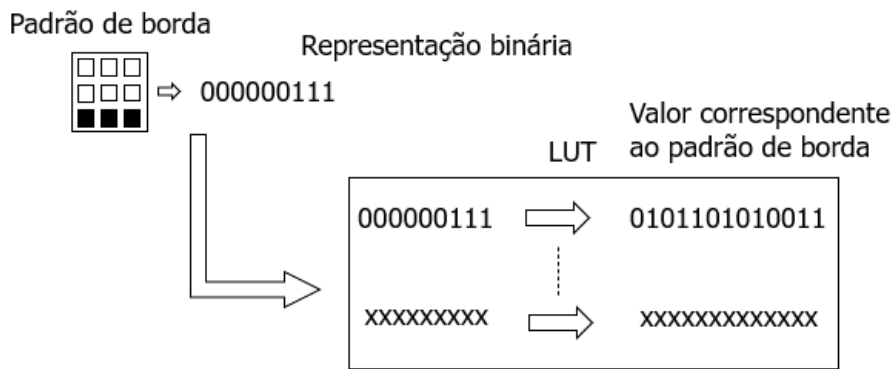
Dado que foi explicado o desenvolvimento do neurônio artificial e da função ativação em FPGA, torna-se importante explicar o desenvolvimento da MLP para extração de bordas. O desenvolvimento da MLP para extração de bordas seguiu de forma diferente ao desenvolvimento do fusor neural.

Anteriormente, foi explicado que o processo de extração de bordas acontece pelo treinamento da MLP para identificar padrões de bordas, estes padrões são representados por matrizes de 9 *pixels*. A MLP é então treinada para identificar estes padrões numa imagem binarizada, e assim é possível destacar as bordas da imagem. Contudo este processo exige computação, ou seja, multiplicação de entradas por pesos, na MLP, e cálculo da função de ativação - processo que aumenta o custo computacional da implementação desta MLP em FPGA.



Analisando os padrões de bordas, que foram apresentados na Figura 7.5, pode-se perceber que cada padrão é representado por um valor único. Além disso, cada padrão pode ser representado por um vetor binário de *pixels*. Desta forma, neste trabalho utilizou-se uma LUT para implementar o extrator de bordas. Na LUT cada vetor binário representa um índice que é mapeado de acordo com seu valor representativo, ou seja, de acordo com o valor de padrão de borda que ele representa - como foi apresentado na Figura 7.5. A Figura 7.13 mostra o esquema para desenvolvimento da LUT usada na extração de bordas.

Figura 7.13 - Desenvolvimento da LUT usada na extração de bordas.



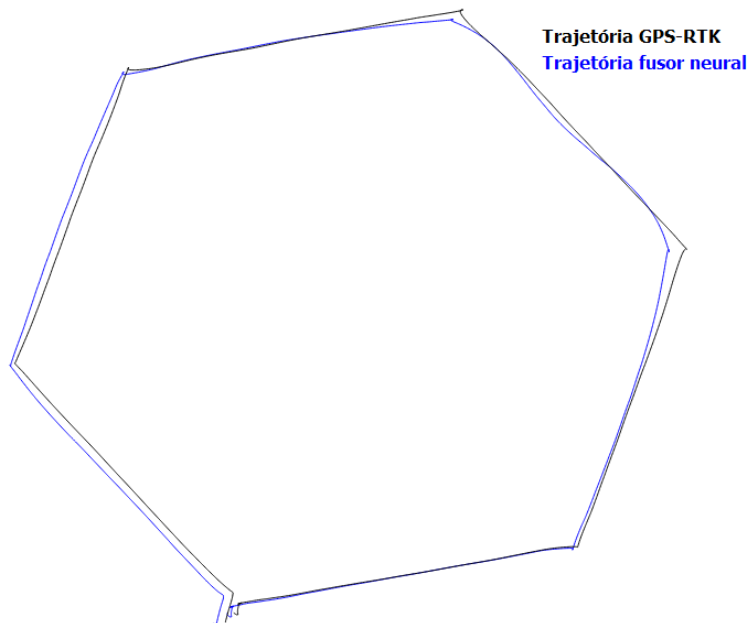
Fonte: Produção do autor.

O processo de extração de bordas utilizando uma LUT é semelhante ao processo de ativação da MLP autoconfigurada para a extração de bordas. A diferença encontra-se no cálculo realizado pela MLP para determinar se um padrão é ou não uma borda. Este cálculo foi substituído por uma consulta à LUT - o que torna o processo mais rápido e com custo computacional menor.

Uma vez abordado o desenvolvimento da MLP autoconfigurada em FPGA, seguiu-se para aplicação do fusor neural. A MLP implementada em FPGA foi descrita anteriormente na Tabela 7.5. Após a implementação em FPGA, a MLP foi ativada novamente e o resultado da ativação, com a trajetória estimada pelo fusor neural é apresentado na Figura 7.14. Na Figura 7.14 são comparadas a trajetória estimada pelo fusor neural com a trajetória medida pelo GPS-RTK. Cabe ressaltar que na Figura 7.14 é apresentado o resultado da ativação para todos os dados usados tanto no treinamento como no teste da MLP autoconfigurada - o objetivo foi ter uma

noção visual do resultado da fusão de dados feita em *hardware* dedicado.

Figura 7.14 - Trajetória estimada pelo fusor neural em comparação com a trajetória medida pelo GPS-RTK.



Em azul está destacado a trajetória estimada pelo fusor neural. Em preto está destacado a trajetória medida pelo equipamento GPS de alta precisão. Esta foi a trajetória para MLP implementada em FPGA na placa *LOGI Pi*.

Fonte: Produção do autor.

Pela Figura 7.14, percebe-se que a trajetória obtida pelo fusão de dados em FPGA se aproximou bastante da trajetória medida pelo GPS-RTK. Cabe ressaltar que, neste caso, não é necessário calcular as incertezas, elas já foram calculadas e mostradas anteriormente na Sessão 7.2, uma vez que nesta etapa não foi utilizada uma MLP nova e sim foi implementado, em FPGA, a MLP obtida pelo MPCA anteriormente.

Apesar do resultado apresentado na Figura 7.14 evidenciar que é possível a fusão de dados em FPGA, com uma MLP, este resultado não é viável para embarque em um VANT. Cabe então explicar porque o resultado não é viável.

O desenvolvimento da MLP em FPGA consiste em programar ou reorganizar as ligações eletrônicas entre os blocos lógicos dentro do dispositivo FPGA (PEDRONI, 2008; PIMENTA, 2017). Assim, cada registrador e LUT utilizados ocupam um espaço dentro do dispositivo. Infelizmente, a MLP não coube completamente dentro do

espaço de blocos lógicos da placa *LOGI Pi* - lembrando que a MLP usada, na fusão de dados, possui duas camadas escondidas. Assim, para poder gerar o resultado da fusão de dados em FPGA foi necessário adotar uma estratégia de computação da fusão de dados em partes.

Como informado anteriormente, a MLP completa não coube dentro da FPGA, mas um neurônio coube, então seguiu-se a estratégia de configurar um neurônio por vez, calcular sua ativação e armazenar o resultado, retornado para a CPU do computador *Raspberry Pi*. Isto foi feito para cada neurônio e cada camada.

Para configurar algo em FPGA, ou seja, para construir um *hardware* em FPGA é necessário enviar para a placa que contém o arranjo de blocos configurável uma arquitetura de *hardware* descrita e compilada em *VHSIC Hardware Description Language* (VHDL) (PEDRONI, 2008; PIMENTA, 2017). Esta arquitetura é, literalmente, montada dentro do dispositivo FPGA e esta montagem, geralmente, é demorada. Assim no contexto da navegação autônoma de VANT, onde a fusão de dados precisa produzir uma estimativa da posição rapidamente, não é possível esperar o tempo de montagem do *hardware* dedicado. Por isso a fusão de dados utilizando o computador *Raspberry Pi* e a placa *LOGI Pi* não é viável.

### 7.3.2 Fusão de dados no computador *Zybo Zynq 7000*

Dado que o utilização do computador *Raspberry Pi*, junto com a placa *LOGI Pi*, não foi viável para implementar a MLP em FPGA, por completo, buscou-se uma outra alternativa que permitisse a implementação do fusor neural em um único dispositivo, sem a necessidade de computar a fusão de dados por partes. A alternativa que se mostrou viável foi utilizar o computador *Zybo Zynq 7000*, também chamado de *Zybo Z7* (BRAGA, 2018; DIGILENT, 2020).

O computador *Zybo Z7*, assim como o *Raspberry Pi*, é um computador *single board*, mas com uma diferença, este computador é construído sobre uma placa que integra sua CPU com uma FPGA da família *Xilinx Zynq-7000* - daí deriva o nome do computador (DIGILENT, 2020).

A família de FPGAs *Xilinx Zynq-7000* é baseada na arquitetura *System on Chip* (SoC). Pode-se entender, de modo simplificado, que a arquitetura SoC consiste na construção e integração dos componentes do computador em um único *chip*, ou seja, no mesmo *chip* estão presentes e integrados a CPU e demais componentes como GPU, USB, circuitos de gerenciamento de energia, etc (DIGILENT, 2020).

A principal vantagem em um computador com arquitetura *SoC* é o seu tamanho (PEDRONI, 2008; PIMENTA, 2017). Um *chip* do tipo SoC, geralmente, é um pouco maior que uma CPU. Entretanto, esta diferença não é expressiva e ele armazena mais componentes, ou seja, mais funcionalidades que uma CPU, mantendo um gasto de energia próximo ou até mesmo menor, quando comparado a CPU integrada a outros componentes externos. Isto significa que o computador *Zybo Z7* não precisa de nenhum outro dispositivo conectado a ele, a FPGA já esta dentro da própria placa e complementamente integrada a CPU o que garante uma comunicação rápida entre a FPGA e a CPU.

As principais características do computador *Zybo Z7* são (DIGILENT, 2020):

- Peso de 114g;
- Comprimento de 12cm e largura de 7,6cm;
- Consumo energético de 12,5wats e 2,5A;
- Processador ARM Cortex 9, dual core com 650MHz e arquitetura de 32bits;
- Memória de 512MB;
- FPGA Xilinx Artix-7;
- Uma porta *Universal Serial Bus* (USB) 2.0;
- Uma saída de vídeo *High-Definition Multimedia Interface* (HDMI);
- Um encaixe para cartão micro-sd.

Como informado, o computador *Zybo Z7* possui integrado à placa uma FPGA do tipo *Xilinx Artix-7*, as principais características dessa FPGA são (BRAGA, 2018):

- 12.800 elementos lógicos;
- 2000 *Configurable Logic Blocks* (CLB);
- 720KB de *Random Access Memory* (RAM).

Percebe-se que as características da FPGA presente no computador *Zybo Z7* são melhores do que as que estão disponíveis na placa *LOGI Pi*, a começar pelo número

de blocos lógicos configuráveis, na placa *LOGI Pi* são 715, já na FPGA do computador *Zybo Z7* são 2.000 blocos lógicos configuráveis, isto significa que há mais espaço na FPGA do computador *Zybo Z7* para a montagem do *hardware* dedicado.

Outra diferença importante entre as FPGAs dos computadores é a forma de comunicação entre a FPGA e a CPU. Especificamente, a FPGA da placa *LOGI Pi* utiliza um barramento de 16bits - os dados são transferidos por 16 pinos que conectavam a FPGA com a CPU. Na FPGA do computador *Zybo Z7*, a comunicação é feita por um barramento do tipo *Advanced Extensible Interface* (AXI) de 32bits. Um barramento do tipo AXI é uma interface de comunicação paralela de alto desempenho, síncrona, de alta frequência, multi-mestre, multi-escravo, projetada principalmente para comunicação SoC.

Dado esta diferença na forma de comunicação entre a FPGA e o computador *Zybo Z7*, foi necessário uma adaptação na construção do neurônio artificial. Esta adaptação consistiu em não dividir o dado em quatro partições binárias, mas em apenas duas.

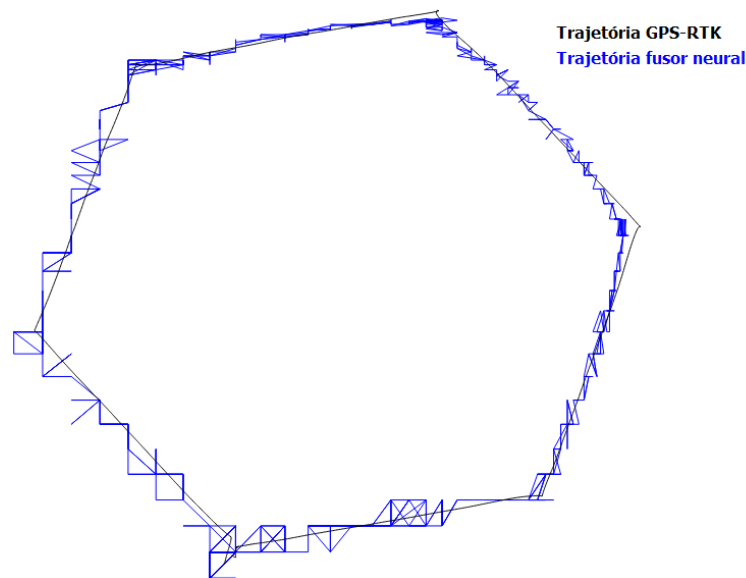
Outra adaptação feita foi diminuir o número da representação em ponto flutuante do dado em pelo menos 4 bits, ou seja, truncar a parte da mantissa em um número. Assim foi possível reduzir o tamanho de bits necessários para enviar um número completo e com isso alguns dados puderam ser enviados dentro da faixa de 32bits.

Uma vez descritos as características e as mudanças necessárias para a implementação do fusor neural, no computador *Zybo Z7*, partiu-se para a implementação da MLP autoconfigurada em FPGA. Desta vez foi possível a construção da MLP por completo e o resultado da estimação da trajetória é apresentado na Figura 7.15. Ressalta-se que do mesmo modo feito na Figura 7.14, são apresentados todos os dados, tanto usados para treinamento como para teste da MLP autoconfigurada e também manteve-se o mesmo padrão de cores, em azul está a estimação da trajetória feita pelo fusor neural e em preto a trajetória real percorrida pelo VANT.

Pela Figura 7.15, percebe-se, claramente, a diferença entre a estimação da trajetória feita pelo fusor neural, implementado em FPGA, e a trajetória medida pelo GPS-RTK. Em grande parte, esta diferença aconteceu por causa do truncamento na mantissa dos dados. Este truncamento foi necessário para poder implementar a MLP por completo em FPGA.

Anteriormente foi informado que as diferenças entre as medidas dos valores das coor-

Figura 7.15 - Trajetória estimada pelo fusor neural implementado no computador *Zybo Z7*.



Em azul está destacado a trajetória estimada pelo fusor neural. Em preto está destacado a trajetória medida pelo equipamento GPS de alta precisão. Esta foi a trajetória para MLP implementada em FPGA no computador *Zybo Z7*, com função de ativação por LUT.

Fonte: Produção do autor.

denadas eram percebidas na mantissa do dado, mais precisamente a partir da quinta casa decimal. Isto é comum para este tipo de dado. Com o truncamento, alguns dados passaram a ter a mesma representação em binário, por exemplo, considerando os valores de latitude 58,494954 e 58,494965, se ocorrer um truncamento, ou seja, um desprezo dos números a partir da quinta casa decimal os valores tornam-se iguais, logo suas representações em binário também serão iguais. Esta representação igual, em binário, fez com que valores que anteriormente representavam índices diferentes ou resultados diferentes na LUT, usada para ativação do neurônio, fossem aproximados como iguais, por isso a estimativa da trajetória, pelo fusor neural, na Figura 7.15 apresenta o aspecto quadriculado, devido a esta aproximação dos valores.

Dado que houve um truncamento, na mantissa dos dados, e que isto provocou diferença nas curvas da trajetória, cabe então calcular esta diferença e mostrar as incertezas a partir da fusão de dados para esta outra implementação em *hardware* dedicado. As incertezas para a estimativa da trajetória, obtidas pela fusão de dados no computador *Zybo Z7*, são mostradas na Tabela 7.6. Cabe informar que estas incertezas foram calculadas sobre os dados usados no conjunto de teste da MLP

autoconfigurada.

Tabela 7.6 - Incertezas para a fusão de dados, com a implementação do fusor neural no computador *Zybo Z7*.

Coordenada	EX12	EX13	Aumento
Erro médio			
Latitude	41,29cm	190,45cm	361,24%
Longitude	111,33cm	215,68cm	93,73%
Desvio			
Latitude	6,38cm	105,92cm	maior que 100%
Longitude	4,17cm	128,03cm	maior que 100%
Variância			
Latitude	40,78cm	11.220,50cm	maior que 100%
Longitude	17,40cm	16.394,13cm	maior que 100%
Erro médio na trajetória			
-	118,89cm	307,88cm	158,96%

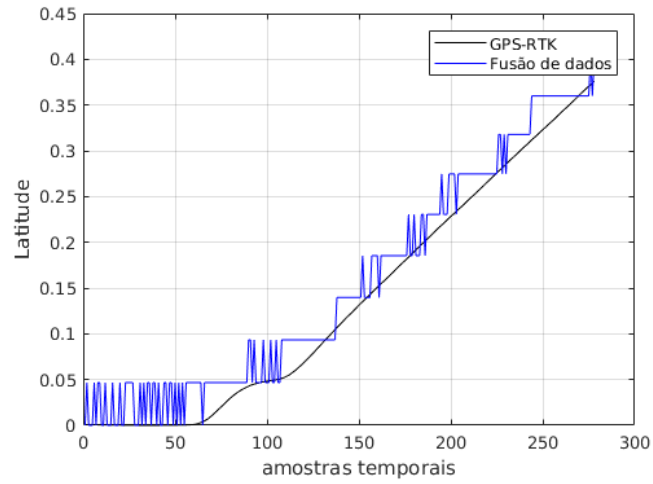
Além das incertezas, para a fusão de dados no computador *Zybo Z7*, a Tabela 7.6 mostra a comparação com as incertezas obtidas com a fusão de dados em CPU. Como este foi o décimo terceiro experimento, as incertezas são mostradas na coluna **EX13** - considerou-se a fusão de dados feita pelo computador *Raspberry Pi* junto da placa *LOGI Pi* o décimo segundo experimento.

A Tabela 7.6 mostra que houve um aumento das incertezas. O erro médio para estimação da latitude foi de 190,45cm e para longitude foi de 215,86cm, o que representa um aumento de 361% e 93% respectivamente, em comparação com a fusão de dados feita em CPU. Claramente, o aumento do erro médio nas coordenadas impactou também no aumento do erro médio na estimação da trajetória, este aumento foi de 158%, o que mostra um grande impacto na fusão de dados quando implementada na FPGA.

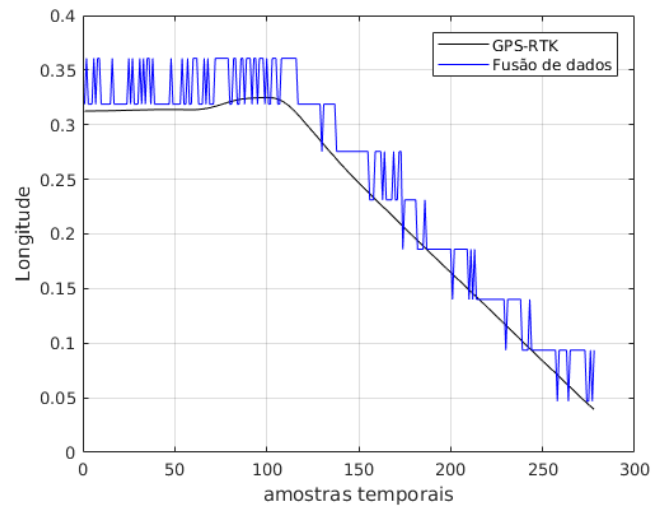
Cabe o destaque para as incertezas que denotam intervalo de confiança, que foram altamente impactadas e tiveram um aumento de mais de 100%. Aponta-se que este aumento foi reflexo do truncamento dos dados.

Uma vez que as incertezas foram calculadas a partir do conjunto de teste, torna-se interessante visualizar as curvas de estimação para cada coordenada obtidas pela fusão de dados. Isto é apresentado na Figura 7.16.

Figura 7.16 - Estimções obtidas pela fusão de dados com FPGA, no computador *Zybo Z7*.



(a)



(b)

(a) Valores para a coordenada de latitude. (b) Valores para a coordenada de longitude.

Fonte: Produção do autor.

Dado que o resultado obtido pela fusão de dados, no computador *Zybo Z7*, mostrou piora em relação a fusão de dados feita em CPU, buscou-se então uma alternativa para melhorar a estimação da trajetória em *hardware* dedicado. A alternativa selecionada foi aplicar técnicas de arquitetura e cálculo baseadas nos algoritmos CORDIC (ANDRAKA, 2001).

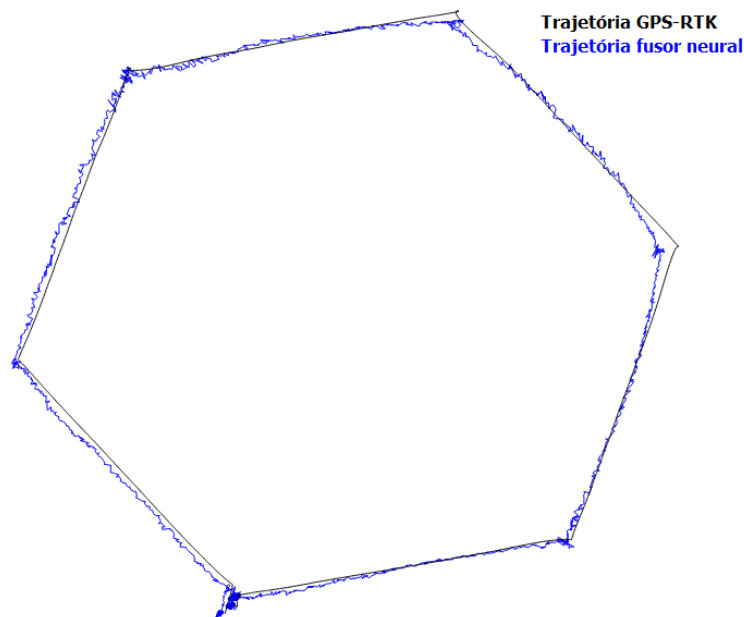
CORDIC consiste de uma coleção de algoritmos e arquiteturas de implementação em



FPGA para calcular uma ampla gama de funções, incluindo funções trigonométricas, funções hiperbólicas, lineares e logarítmicas (ANDRAKA, 2001). O nome CORDIC é um acrônimo para *Coordinate Rotation Digital Computer*.

O que piorou as incertezas da fusão de dados e também provocou a formato quadriculado na estimação da trajetória, mostrada na Figura 7.16, foi o truncamento dos dados juntamente com a função de ativação do neurônio, implementada em LUT. Para melhorar este resultado substituiu-se a LUT por uma implementação da função sigmoid, de acordo com as definições dos algoritmos CORDIC (ANDRAKA, 2001). Com isso, obteve-se uma nova implementação em FPGA e também uma nova estimação da trajetória. Esta nova estimação da trajetória é mostrada na Figura 7.17.

Figura 7.17 - Trajetória estimada pelo fusor neural implementado no computador *Zybo Z7*, com ativação CORDIC.



Em azul está destacado a trajetória estimada pelo fusor neural. Em preto está destacado a trajetória medida pelo equipamento GPS de alta precisão. Esta foi a trajetória para MLP implementa em FPGA no computador *Zybo Z7*, com função de ativação por CORDIC.

Fonte: Produção do autor.

Percebe-se pela Figura 7.17, que houve uma melhora significativa na estimação da trajetória quando se compara com a imagem mostrada, anteriormente, na Fi-

gura 7.15. Dado que houve melhora na estimação da trajetória torna-se importante mostrar as incertezas para esta nova fusão de dados. Estas incertezas são mostradas na Tabela 7.7.

Tabela 7.7 - Incertezas para a fusão de dados, com a implementação do fusor neural no computador *Zybo Z7* e ativação CORDIC.

Coordenada	EX12	EX14	Diferença
Erro médio			
Latitude	41,29cm	55,00cm	33,20%
Longitude	111,33cm	103,86cm	-6,7%
Desvio			
Latitude	6,38cm	31,67cm	maior que 100%
Longitude	4,17cm	42,99cm	maior que 100%
Variância			
Latitude	40,78cm	1.003,29cm	maior que 100%
Longitude	17,40cm	1.848,88cm	maior que 100%
Erro médio na trajetória			
-	118,89cm	109,50cm	-7,89%

A Tabela 7.7 também faz uma comparação com as incertezas da fusão de dados feita em CPU e como este foi o décimo quarto experimento, as incertezas são mostradas na coluna **EX14**. Na coluna denominada diferença os valores positivos significam aumento da incerteza e o valores negativos redução.

O erro médio da estimação da trajetória, quando usado a arquitetura CORDIC para implementação da função de ativação, foi melhorado. Inclusive houve uma redução neste erro, que ficou 7% menor. Esta melhora aconteceu devido uma redução no erro médio da estimação da coordenada longitude. Cabe ressaltar que esta melhora não advém de alguma alteração na topologia da MLP implementada em FPGA, a melhora é decorrente da forma como a arquitetura CORDIC modificou o resultado da função de ativação do neurônio, em outras palavras, a implementação da função de ativação, seguindo a arquitetura CORDIC, provocou algumas variações nas curvas de estimação, estas variações alteram o erro médio na estimação das coordenadas.

É importante ressaltar que as incertezas, que denotam intervalo de confiança, também foram melhoradas, quando comparadas as incertezas do experimento **EX13**, mostradas anteriormente na Tabela 7.6. Mas, cabe evidenciar que estas incertezas continuam mais altas do que as incertezas obtidas com a fusão de dados em CPU. Contudo, pode-se concluir que entre usar a função de ativação, do neurônio, baseada

em LUT, como feito anteriormente, ou usar o padrão CORDIC, para implementação da função de ativação do neurônio, em FPGA, a arquitetura CORDIC é a melhor opção.

Assim como no experimento anterior, uma vez que as incertezas foram calculadas a partir do conjunto de teste, torna-se interessante visualizar as curvas de estimação para cada coordenada obtidas pela fusão de dados. Isto é apresentado na Figura 7.18.

Apesar da arquitetura CORDIC se sobressair em relação a LUT, a complexidade da construção da função de ativação usando CORDIC é maior. Isto significa que usar CORDIC para construção do *hardware* dedicado consome mais espaço dentro da FPGA.

Como apresentado anteriormente, o espaço gasto para construção do *hardware* dedicado, ou seja, a quantidade de blocos lógicos configuráveis usados, é um insumo que precisa ser economizado, sob pena de não ser possível implementar uma MLP em FPGA caso não se tenha espaço suficiente na FPGA, ou seja, quanto mais simples o *hardware* for, menos blocos lógicos configuráveis são necessários para sua construção em FPGA.

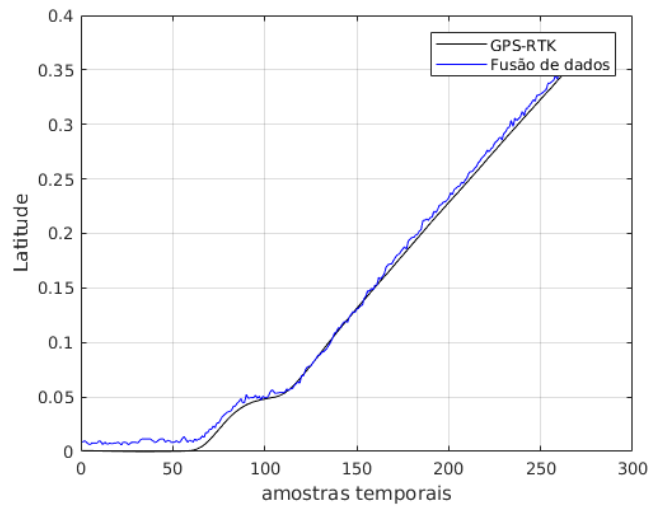
Partindo do princípio de economizar espaço dentro da FPGA, uma vez que a LUT é mais simples que a arquitetura CORDIC, buscou-se investigar se é possível melhorar a implementação da função de ativação do neurônio com uma LUT.

Como informado anteriormente, o problema com a LUT aconteceu porque valores truncados, com representação binária igual, foram juntos na mesma resposta para a ativação, ou seja, representavam o mesmo índice na LUT que foi usada como função de ativação e conseqüentemente foram relacionados ao mesmo valor de ativação. Desta forma, buscou-se então desenvolver a LUT seguindo um novo esquema, de forma a evitar relacionar mais de um índice ao mesmo valor de ativação.

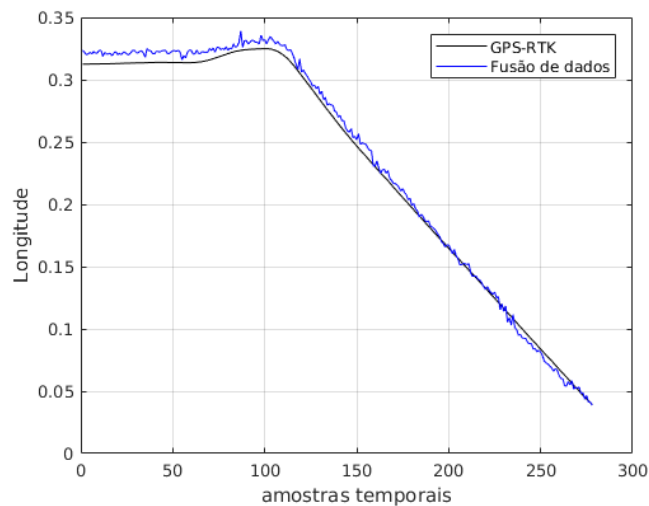
A novo esquema para construção da LUT consiste em definir índices principais e seus respectivos valores de ativação juntamente com índices secundários e valores de ativação interpolados. Estes índices principais correspondem, por exemplo, aos índices que definiram a forma quadriculada da estimação da trajetória mostrada, anteriormente, na Figura 7.15.

Os índices restantes, ou seja, os que não são definidos como principais, outrora foram relacionados aos valores de ativação dos índices principais, agora, passam a ser relacionados a valores próprios de ativação. Este valores próprios de ativação corres-

Figura 7.18 - Estimacões obtidas pela fusão de dados com FPGA, no computador *Zybo Z7*, com ativação CORDIC.



(a)



(b)

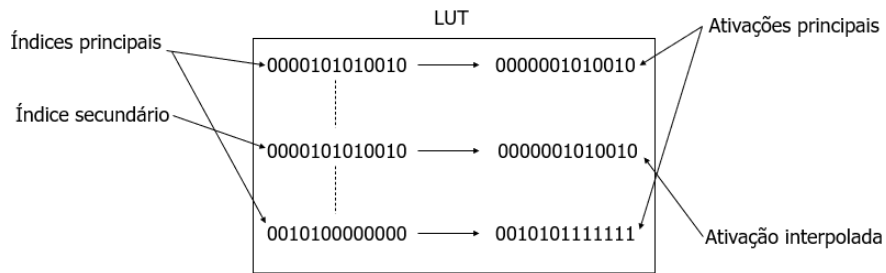
(a) Valores para a coordenada de latitude. (b) Valores para a coordenada de longitude.

Fonte: Produção do autor.

pondem a valores interpolados entre os valores de ativação relacionados aos índices principais, em outras palavras, considera-se um índice como um valor de entrada para a LUT, quando este valor de entrada está entre dois valores principais de entrada, a saída da LUT será uma interpolação linear entre as duas saídas relacionadas aos valores principais de entrada.

Para facilitar o entendimento deste novo esquema a Figura 7.19 ilustra como deve ser computada a saída da LUT que representa a função de ativação - na Figura 7.19, os valores binários são apenas ilustrativos, eles não representam os valores reais usados na construção da LUT.

Figura 7.19 - Desenvolvimento da LUT com ativação interpolada.



Fonte: Produção do autor.

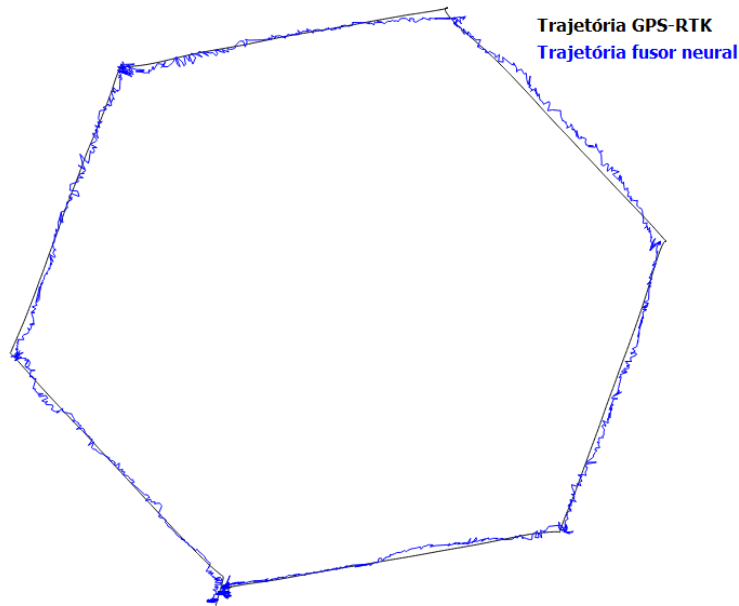
Uma vez definido o novo esquema para construção da LUT, partiu-se para uma nova implementação do fusor neural, que gerou um novo resultado. A trajetória estimada para esta fusão de dados é mostrada na Figura 7.20.

Pela Figura 7.20, é possível perceber que este novo esquema de construção da LUT gerou um resultado para fusão de dados que é semelhante a arquitetura CORDIC. Inclusive, teve uma aproximação melhor em relação a trajetória medida pelo GPS-RTK, quando comparado ao resultado da fusão de dados obtido com a primeira implementação de LUT - resultado apresentado, anteriormente, na Figura 7.15.

Dado o resultado apresentado na Figura 7.20, torna-se importante mostrar as incertezas para este experimento. Estas incertezas são mostradas na Tabela 7.8, que também mostra a comparação das incertezas obtidas neste experimento com as incertezas obtidas pela fusão de dados em CPU. Como este foi o décimo quinto experimento, as incertezas estão mostradas na coluna **EX15**

Na Tabela 7.8, na coluna denominada de diferença, os valores positivos representam aumento da incerteza e os valores negativos redução. Houve uma redução do erro médio na estimação da trajetória, redução menor ainda do que quando se usou a arquitetura CORDIC. Esta redução foi provocada por uma redução no erro médio de estimação da coordenada longitude, contudo, cabe evidenciar que a redução não

Figura 7.20 - Trajetória estimada pelo fusor neural implementado no computador *Zybo Z7* e ativação LUT interpolada.



Em azul está destacado a trajetória estimada pelo fusor neural. Em preto está destacado a trajetória medida pelo equipamento GPS de alta precisão. Está foi a trajetória para MLP implementa em FPGA no computador *Zybo Z7*, com função de ativação por LUT interpolada.

Fonte: Produção do autor.

Tabela 7.8 - Incertezas para a fusão de dados, com a implementação do fusor neural no computador *Zybo Z7* e ativação LUT com interpolação.

Coordenada	EX12	EX15	Diferença
Erro médio			
Latitude	41,29cm	63,96cm	54,90%
Longitude	111,33cm	47,57cm	-57,27%
Desvio			
Latitude	6,38cm	38,97cm	maior que 100%
Longitude	4,17cm	36,98cm	maior que 100%
Variância			
Latitude	40,78cm	1.518,83cm	maior que 100%
Longitude	17,40cm	1.367,69cm	maior que 100%
Erro médio na trajetória			
-	118,89cm	88,70cm	-25,39%

foi provocada por alguma alteração na topologia da MLP implementada em FPGA. A redução é decorrente da forma como a novo esquema de implementação da LUT calcula a ativação do neurônio.

Assim como nos experimentos anteriores, uma vez que as incertezas foram calculadas a partir do conjunto de teste, torna-se interessante visualizar as curvas de estimação para cada coordenada obtidas pela fusão de dados. Isto é apresentado na Figura 7.21.

### 7.3.3 Considerações sobre o desenvolvimento do coprocessador em *hardware* dedicado

Nesta etapa, o foco está voltado para investigar o desempenho ou impacto da construção da MLP autoconfigurada em *hardware* dedicado. Isto foi realizado com o teste de três implementações do fusor neural em FPGA - variando a forma como a função de ativação do neurônio é construída.

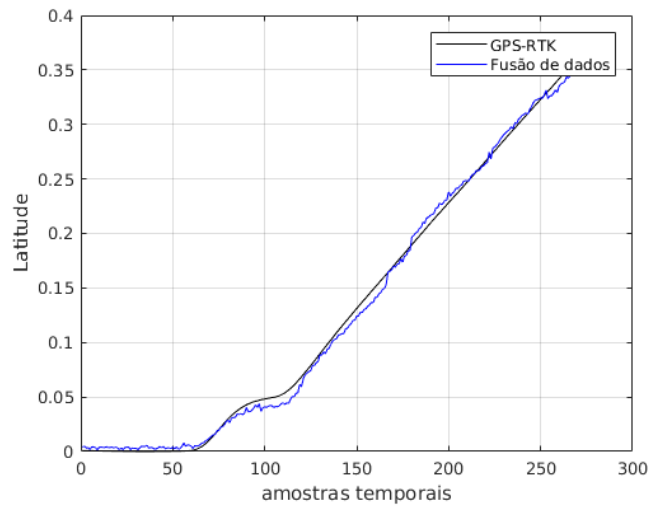
As investigações culminaram numa versão melhorada de LUT, que permitiu um resultado ótimo, junto com uma menor complexidade de desenvolvimento em FPGA. Desta forma, trata-se de um *hardware* mais simples capaz de fusionar os dados coletados de um VANT em voo e chegar a uma estimação da posição da aeronave próxima a posição obtida por um GPS-RTK.

Cabe ressaltar que as implementações tanto com CORDIC, quanto com a LUT interpolada resultaram em valores menores para o erro médio de estimação da trajetória. Contudo, isto não era esperado e tampouco foi pensado de maneira deliberada. Como comprovação disto, percebe-se que os valores das incertezas que denotam intervalo de confiança estão mais altos que os obtidos com a CPU.

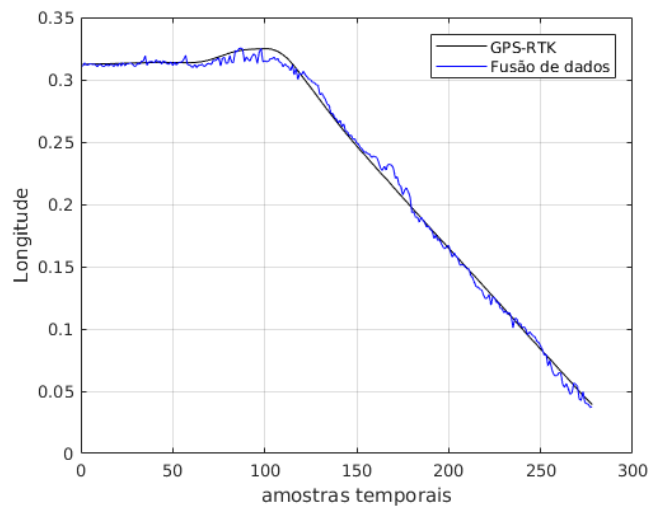
Apesar das incertezas obtidas pela fusão de dados em FPGA serem maiores do que as obtidas pela fusão de dados em CPU, entende-se que o aumento das incertezas é consequência da variação nas curvas de estimatórias das coordenadas latitude e longitude. Mesmo com o aumento das incertezas, nos casos onde foram utilizados a arquitetura CORDIC e a LUT interpolada, foi possível constatar a proximidade da estimação da trajetória por fusão de dados, quando comparada a trajetória medida pelo GPS-RTK.

É importante lembrar que, em vários momentos, foi informado o objetivo de se ter uma MLP com arquitetura, topologia, otimizada. Nesta etapa, os experimentos justificaram a aplicação do MPCA. Além disso cabe informar que o MPCA foi escolhido por ser um algoritmo pensado por brasileiros e desenvolvido a partir de

Figura 7.21 - Estimacões obtidas pela fusão de dados com FPGA, no computador *Zybo Z7* e ativação LUT interpolada.



(a)



(b)

(a) Valores para a coordenada de latitude. (b) Valores para a coordenada de longitude.

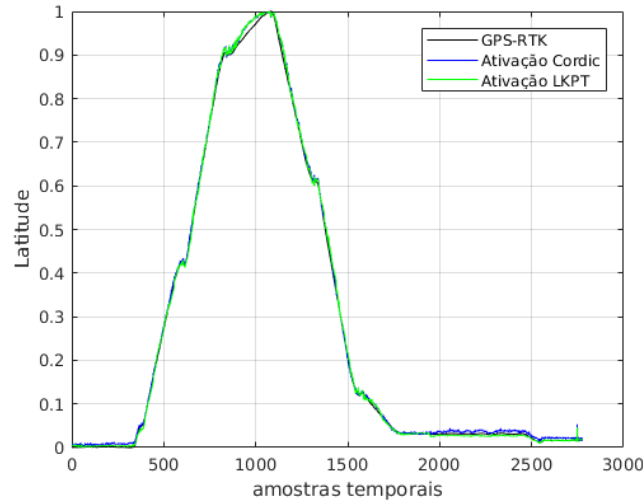
Fonte: Produção do autor.

pesquisas feitas no INPE - neste trabalho buscou-se prestigiar a tecnologia brasileira.

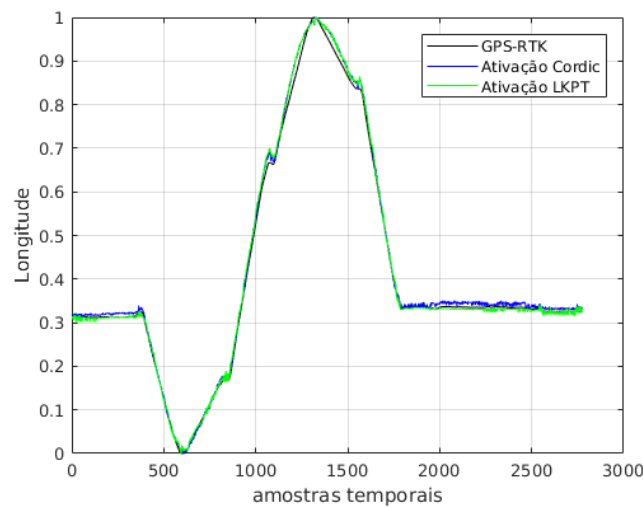
Para se ter uma comparação visual da estimacão da coordenada latitude, é apresentado na Figura 7.22 a estimacão feita pelas formas de ativação CORDIC e LUT interpolada. Também na Figura 7.22 é apresentado a estimacão para coordenada longitude.



Figura 7.22 - Estimacões para cada coordenada obtidas pelas ativações CORDIC e LUT interpolada.



(a)



(b)

(a) Valores para a coordenada de latitude. (b) Valores para a coordenada de longitude. Na cor preta, esto os valores medidos pelo GPS-RTK, Na cor azul, esto os valores obtidos, pela fuso de dados, com ativao CORDIC e na cor verde, esto os valores obtidos, pela fuso de dados, com ativao LUT.

Fonte: Produo do autor.



## 8 QUANTIFICAÇÃO DE INCERTEZAS

Nos capítulos anteriores, em cada experimento foi mostrado as incertezas para a estimação da trajetória por fusão de dados. Isto é importante à medida que são elas que indicam o intervalo de confiança que a metodologia alcançou. Ressalta-se que o intervalo de confiança é usado para indicar a confiabilidade da estimação da trajetória.

Diante desta funcionalidade, é interessante que estas informações sobre intervalo de confiança estejam disponíveis durante o voo. Isto auxiliaria, por exemplo, a tomada de decisão de um controlador de voo em aceitar ou não a estimação feita por fusão de dados.

Neste capítulo, são investigadas formas de disponibilizar incertezas para aceitar ou não a estimação feita por fusão de dados ou ainda para corrigir a posição do VANT, caso a fusão de dados não seja apropriada ou não possa ser executada.

### 8.1 Estimação do desvio padrão

Nos experimentos anteriores, as incertezas eram calculadas sempre sobre um conjunto de teste, que foi denominado de conjunto de generalização. Ele foi previamente separado e é uma parte de um conjunto de dados obtido após a realização de um voo.

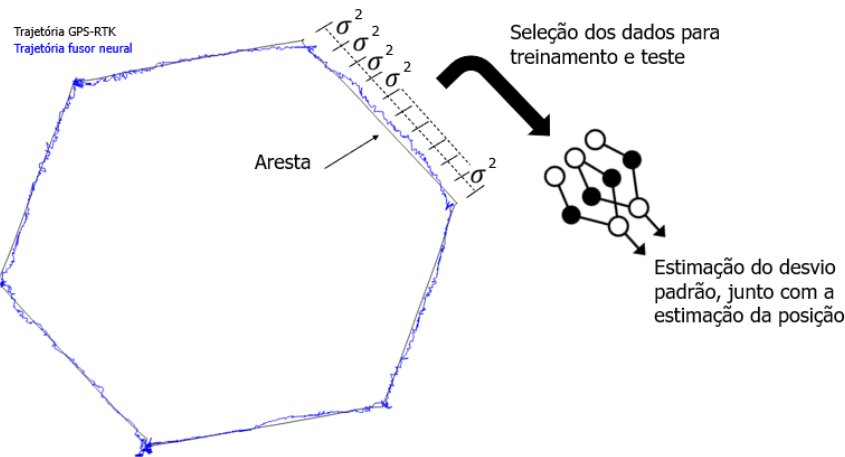
Contudo, durante um voo, não há a disponibilidade de um conjunto de teste. Desta forma, não há como estimar um erro médio para trajetória da mesma forma que foi feito nos experimentos anteriores.

Uma das incertezas que indicam intervalo de confiança para a fusão de dados é o desvio padrão, que é dependente do erro médio de estimação da trajetória. Assim, torna-se importante disponibilizar o desvio padrão associado ao erro médio de estimação da trajetória por fusão de dados, isto possibilita ter um índice que expresse a confiabilidade da estimação da trajetória.

A bordagem desenvolvida para obter o desvio padrão, consistiu em dividir a trajetória de voo, usada nos experimentos do Capítulo 7, em partes. Cada parte representa uma aresta na figura hexagonal que descreve a trajetória percorrida pelo VANT. Cada aresta foi particionada em retas de tamanhos iguais, ou seja, em séries temporais de tamanhos iguais. Então para cada série temporal foi calculado um desvio padrão. Isto gerou um conjunto de desvios associados a cada aresta da trajetória.

Todos os conjuntos de desvio padrão, juntos, formaram uma base de dados que foi usada para treinar uma MLP autoconfigurada, para fornecer um desvio padrão durante o voo. A Figura 8.1 ilustra a abordagem desenvolvida para estimar o desvio padrão durante o voo.

Figura 8.1 - Separação dos índices do desvio padrão por aresta, na trajetória do VANT.



Fonte: Produção do autor.

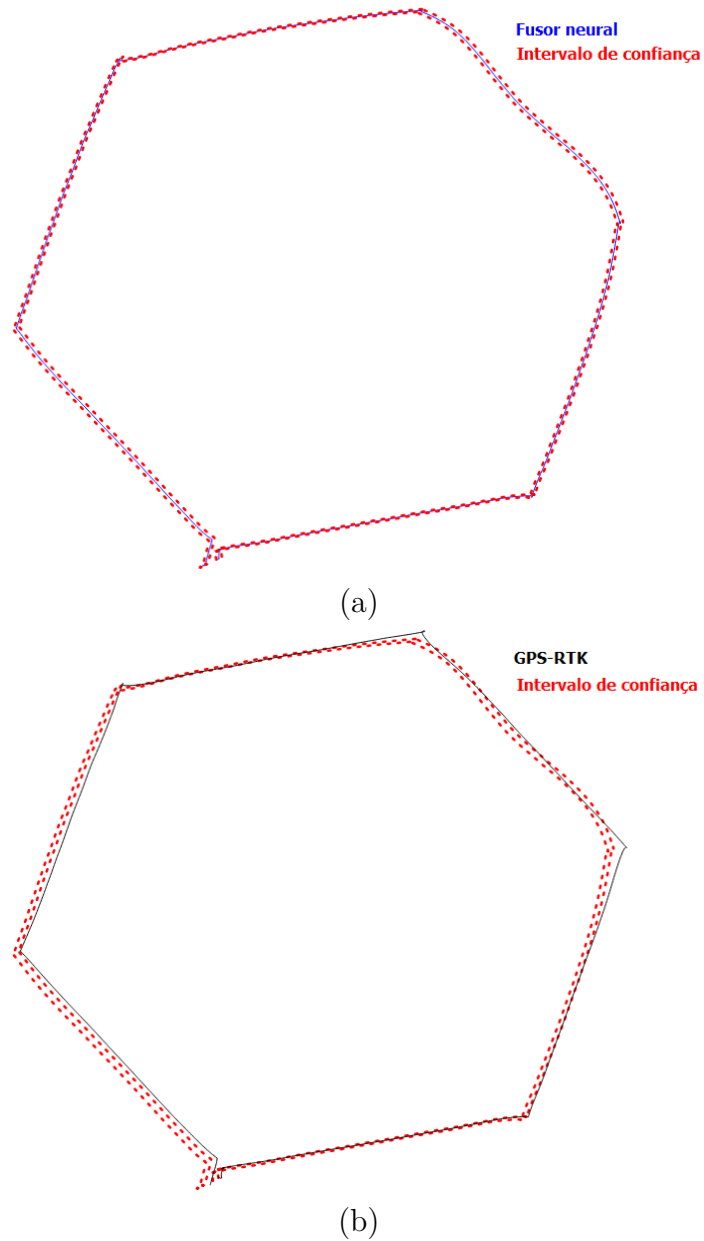
Cabe informar que, para este experimento, a base de dados que contém o conjunto de desvios foi unida aos dados usados para treinamento da MLP na estimativa da trajetória por fusão de dados. Desta forma, o novo treinamento resultou em uma MLP que, juntamente da estimativa da posição do VANT, também informa um desvio associado a esta estimativa. Aponta-se também que a MLP obtida neste experimento não foi implementada em FPGA. Portanto, o resultado para quantificação desta incerteza foi obtido com ativação da MLP em CPU.

Para a implementação da MLP, neste experimento, também foi utilizado o MPCA. Então, seguiu-se o mesmo procedimento de particionamento dos conjuntos de treinamento, validação e generalização, como realizado anteriormente.

Uma vez que se tenha a MLP para fornecer o intervalo de confiança, torna-se importante mostrar o intervalo fornecido para a estimativa da trajetória, que é mostrado na Figura 8.2.

Na Figura 8.2, o intervalo de confiança está denotado pelas linhas vermelhas, tra-

Figura 8.2 - Intervalo de confiança para a estimação da trajetória.



(a) Intervalo de confiança junto com a estimação da trajetória pelo fusor neural. (b) Intervalo de confiança junto da trajetória medida pela GPS-RTK.

Fonte: Produção do autor.

cejadas. Portanto, são mostradas duas imagens, em que a primeira é o intervalo de confiança sobreposto a estimação da trajetória obtida pelo fusor neural, já a segunda é o intervalo de confiança sobreposto a trajetória medida pelo GPS-RTK.

Pela Figura 8.2, é possível ter uma impressão sobre a confiabilidade da estimação

da trajetória, obtida pelo fusor neural. A Figura 8.2 mostra uma faixa de intervalo de confiança, denotada por dois limites, que definem o espaço onde a estimação da posição do VANT pode ocorrer. Estes limites mostram, claramente, que o intervalo de confiança da estimação da trajetória sempre se aproxima da trajetória percorrida pela aeronave.

Dado que são dois limites para descrever o intervalo de confiança, torna-se interessante mostrar o erro médio entre cada limite e a trajetória percorrida pelo VANT. Este erro médio é mostrado na Tabela 8.1. Os erros médios mostrados na Tabela 8.1 foram calculados pela diferença entre os limites do intervalo de confiança e a trajetória medida pelo GPS-RTK.

Tabela 8.1 - Erros médios para a faixa de intervalo de confiança.

Coordenada	Limite superior	Limite inferior
Erro médio		
Latitude	25,68cm	77,89cm
Longitude	83,14cm	181,49cm
Erro médio na trajetória		
-	91,04cm	200,41cm

## 8.2 Estimação da trajetória por filtro de partículas

Na literatura, encontram-se fontes que mostram a aplicação de técnicas de fusão de dados para problemas relacionados a robótica, principalmente os problemas em relação ao deslocamento autônomo de robôs e estimação da posição no espaço de deslocamento (KHALEGHI et al., 2013; MOHD et al., 2017; BRAGA, 2018; PAULINO, 2019). Em particular, destaca-se uma técnica denominada de Filtro de partículas (FP), também chamado de Método Sequencial de Monte Carlo (MSMC) (SCHOTT et al., 2017; BRAGA, 2018; PETUKHOV et al., 2019; GUANG; JI, 2020).

O FP é uma técnica baseada em filtragem bayesiana ou estimação bayesiana, que já foi aplicada à estimação da posição de VANT por visão computacional, por isso, aplicou-se também neste trabalho o FP (BRAGA, 2018). Contudo, sua aplicação ocorreu sobre os dados inerciais. Dado que o FP consiste de uma técnica baseada em estimação bayesiana, cabe informar que ela é uma técnica probabilística que objetiva aquilatar uma Função Densidade de Probabilidade (FDP) (FURTADO, 2008; PETUKHOV et al., 2019; GUANG; JI, 2020).

Considerando o contexto do deslocamento autônomo, o processo para encontrar a FDP usa parâmetros (dados) separados em duas categorias, que são os parâmetros relacionados as medidas observacionais e um modelo matemático. Este modelo matemático pode ser entendido como uma representação sistêmica de características relacionadas ao deslocamento (BRAGA, 2018).

Para navegação autônoma de VANT, as medidas observacionais correspondem aos valores obtidos pelo INS e GNSS, por exemplo. O modelo matemático representa alguma técnica, que foi construída para que se consiga, computacionalmente, simular o comportamento da aeronave, ou seja, simular como ela funciona. Isto possibilita a aproximação das medidas observacionais ou de alguma medida considerada como correta, o que geralmente é avaliado por um operador de verossimilhança (FURTADO, 2008; PETUKHOV et al., 2019; GUANG; JI, 2020).

Dado que o modelo matemático é uma representação de um sistema e um sistema consiste de um conjunto de objetos reunidos sob alguma forma de interação, diz-se que há relação de causa e efeito nos fenômenos que ocorrem com esses objetos. Quando aspectos dos objetos do modelo variam no tempo o modelo é classificado como dinâmico (BRAGA, 2018).

Na literatura, é dito que, para estimação bayesiana, há necessidade de se ter um modelo adequado, o que é definido como algo que garante o objetivo para o qual foi proposto. Ele descreve, de modo essencial e simples, o conjunto de objetos que possui e as suas interações, para que seja possível a construção do modelo em forma de um algoritmo (FURTADO, 2008).

A variação dos aspectos do modelo dinâmico no tempo, resultam em aplicar a este modelo um valor, um estado. Por analogia, para navegação autônoma de VANT, o estado do modelo durante a trajetória seria uma composição dos valores obtidos pelos sensores de atitude. Em outras palavras, um vetor com informações das coordenadas, dos ângulos de Euler e das taxas de aceleração.

Cabe ressaltar que, no experimento realizado nesta etapa, não foi possível capturar medidas infinitesimais, ou seja, as observações sobre cada estado são obtidas em tempo discreto. Portanto, considera-se que o modelo usado nesta etapa é de tempo discreto.

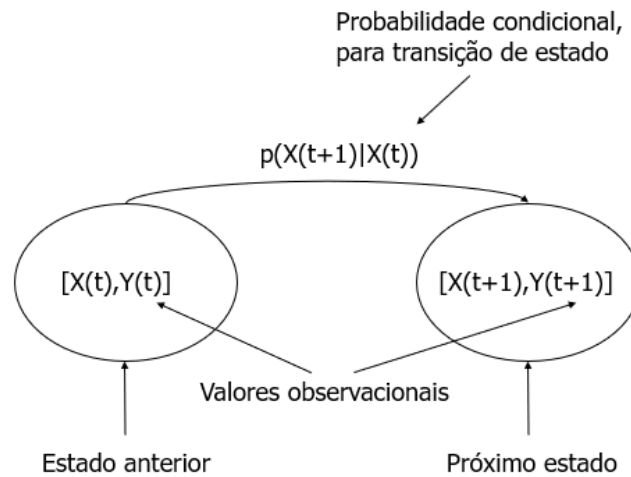
Para modelos de tempo discreto, é possível aplicar a propriedade de Markov, se e somente se for identificado que, para o conjunto de estados de um modelo, um estado

atual depende somente de seu estado anterior, ou seja, a passagem, do modelo, para um determinado estado depende somente do estado anterior. Esta propriedade pode ser representada como na Equação 8.1:

$$p(x_{t+1}|x_0, \dots, x_t) = p(x_{t+1}|x_t) \quad (8.1)$$

Na Equação 8.1, define-se que a probabilidade de passagem do modelo para um estado  $x_{t+1}$  é dependente, somente, do estado  $x_t$ , sendo  $t$  o tempo. A Figura 8.3 ilustra este processo.

Figura 8.3 - Transferência de estados por probabilidade condicional.



Fonte: Produção do autor.

Na Figura 8.3, são mostrados dois estados apenas, mas é comum modelos terem vários estados. Considerando um modelo com  $k$  estados, onde  $k = 1, \dots, t + 1$ ,  $x_k$  é considerado um vetor que contém os valores relacionados ao estado e  $y_k$  é considerado um vetor que contém os valores observacionais do estado.

Na estimação bayesiana, os aspectos relacionados a  $x_k$  e  $y_k$  podem variar no tempo, o que faz com que eles sejam considerados variáveis aleatórias que podem ser representados com uma FDP. Então, elas são representadas pela probabilidade condicional  $p(x_t|y_t)$ , ou seja, a FDP significa a probabilidade de estar num determinado estado, dado que foram percebidos determinados valores observacionais (CAMPOS VELHO;



FURTADO, 2011; GUANG; JI, 2020).

No modelo de espaço de estados dinâmico, a probabilidade condicional  $p(x_t|y_t)$  é calculada de forma recursiva, a cada mudança de tempo e a cada alteração nos valores observacionais (CAMPOS VELHO; FURTADO, 2011; BRAGA, 2018; GUANG; JI, 2020). A partir da probabilidade condicional e se a dinâmica do modelo for conhecida é possível calcular a probabilidade de transição entre os estados, que foi mostrada na Equação 8.1. O FP é uma técnica que busca estimar de forma recursiva a probabilidade condicional  $p(x_t|y_t)$  (CAMPOS VELHO; FURTADO, 2011; PETUKHOV et al., 2019; GUANG; JI, 2020). Existem variações de implementações do FP, neste experimento utilizou-se uma implementação baseada em um FP não-extensivo (BRAGA, 2018).

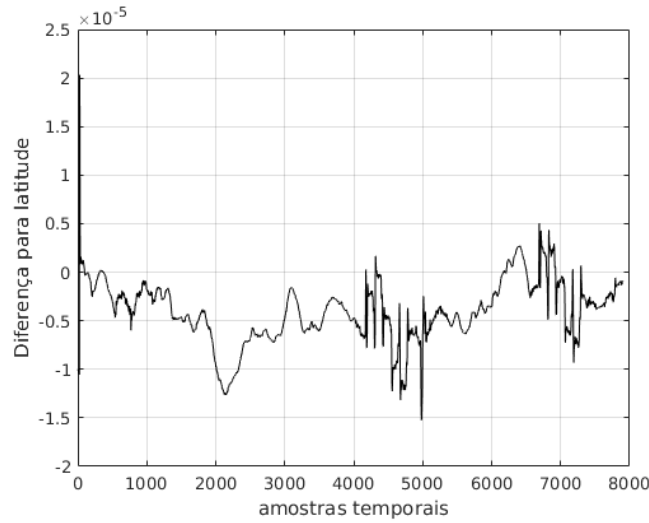
Como informado, anteriormente, é possível calcular a probabilidade de transição entre os estados do modelo, se a dinâmica do modelo é conhecida. Contudo, neste trabalho, não se tem a dinâmica referente aos VANTs utilizados. Por isso, seguiu-se com uma abordagem diferente, onde a dinâmica foi baseada no erro (diferença) e desvio padrão, obtidos entre as medidas do GPS-RTK e do INS, para cada coordenada. Cabe informar que este experimento utilizou apenas os dados do VANT disponível no IEAv - VANT usado nos experimentos do Capítulo 7.

A abordagem para construção do FP, utilizado neste experimento, consiste em alguns passos. Para facilitar a explicação da construção e aplicação do FP, doravante, adota-se que a denominação modelo representa o processo de deslocamento do VANT no espaço e que dados observacionais representam os valores, de cada coordenada, obtidos pelo INS, num dado instante de tempo.

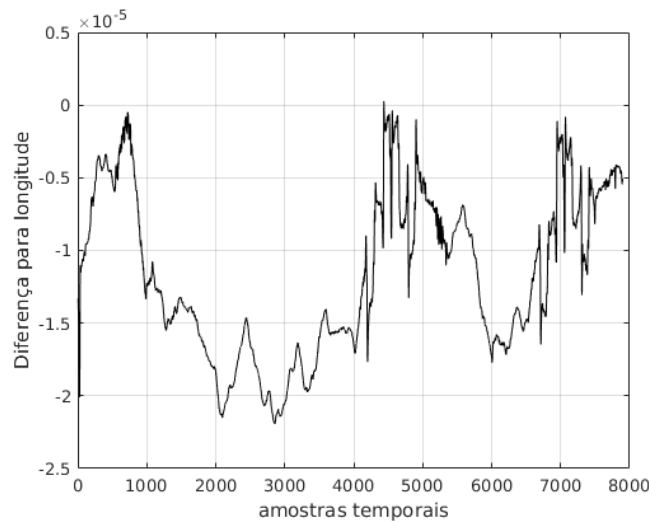
O primeiro passo para construção do FP é calcular o erro, ou seja, a diferença entre as medidas obtidas pelo INS e pelo GPS-RTK embarcados, o que é feito para cada coordenada. Como resultado, tem-se uma série temporal da diferença. Esta diferença é mostrada na Figura 8.4.

O Segundo passo consiste em encontrar uma distribuição de probabilidade que defina a série temporal da diferença, de cada coordenada. Neste experimento, utilizou-se distribuições Gaussianas. A partir das distribuições Gaussianas, tem-se os seus respectivos valores de desvio. Estas distribuições foram aplicadas na etapa de amostragem (espalhamento) do FP. As distribuições de cada coordenada são mostradas na Figura 8.5.

Figura 8.4 - Diferença entre as medidas do GPS-RTK e INS.



(a)



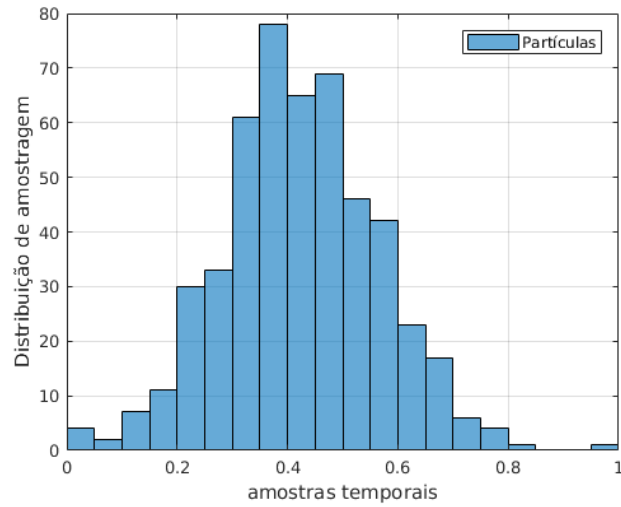
(b)

(a) Valores para a coordenada latitude. (b) Valores para a coordenada longitude.

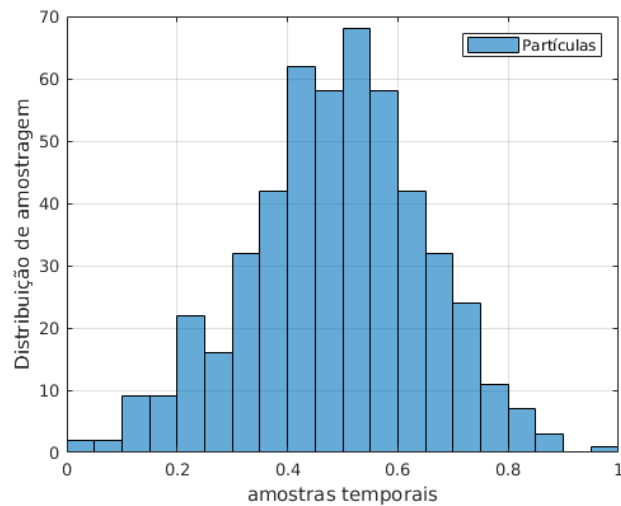
Fonte: Produção do autor.

Uma vez que se tenha a distribuição de amostragem, para um determinado estado do modelo, provoca-se perturbações sobre os aspectos que descrevem este estado, que no caso da navegação do VANT são os valores que denotam sua localização no espaço, as coordenadas latitude e longitude. Estas perturbações são geradas considerando a distribuição de amostragem e resultam em vários novos estados, que no algoritmo do FP recebem a denominação de partículas (CAMPOS VELHO; FURTADO, 2011). Cabe informar que a quantidade de partículas a ser gerada é um dos parâmetros a

Figura 8.5 - Distribuições para amostragem no FP.



(a)



(b)

(a) Valores para a coordenada latitude. (b) Valores para a coordenada longitude.

Fonte: Produção do autor.

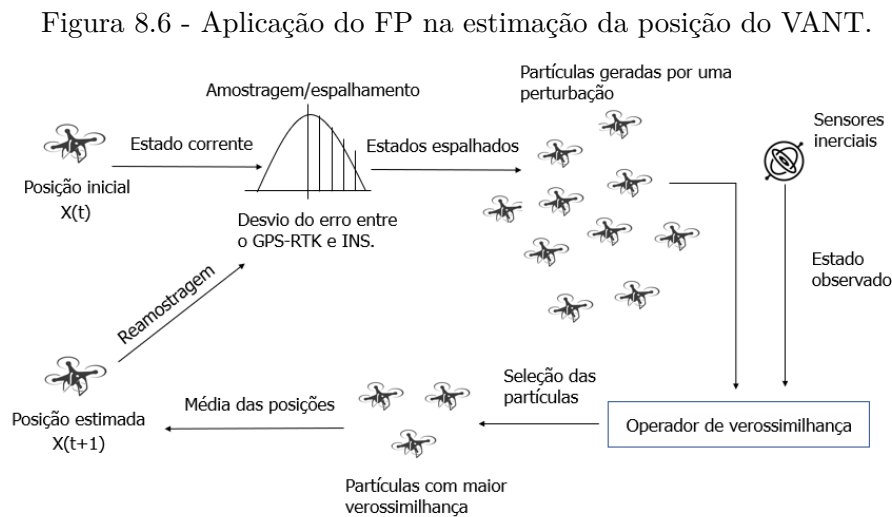
ser definidos no algoritmo do FP.

De posse do conjunto de partículas, aplica-se este conjunto, juntamente com os dados observacionais à um operador de verossimelhança (CAMPOS VELHO; FURTADO, 2011). O operador de verossimelhança consiste em uma função densidade de probabilidade, que fornece uma indicação de quanto o modelo adotado descreve o funcionamento do sistema, ou seja, do VANT no mundo real. O mundo real é representado

pelos dados observacionais (FURTADO, 2008). O operador de verossimilhança classifica as partículas, indicando quais partículas melhor representam a dinâmica da aeronave.

A partir da classificação das partículas, escolhe-se as que ficaram melhor classificadas. A estimativa final do estado do modelo, ou seja, da posição do VANT é obtida pela média dos aspectos que definem o estado de cada uma das partículas escolhidas. Em outras palavras, a posição estimada para o VANT é definida pela média dos valores das coordenadas de latitude e longitude, que denotam a localização de cada partícula no espaço.

O processo de aplicação do FP é feito de maneira recursiva, durante a trajetória do VANT. A Figura 8.6 ilustra a construção e aplicação do FP, utilizado neste experimento.

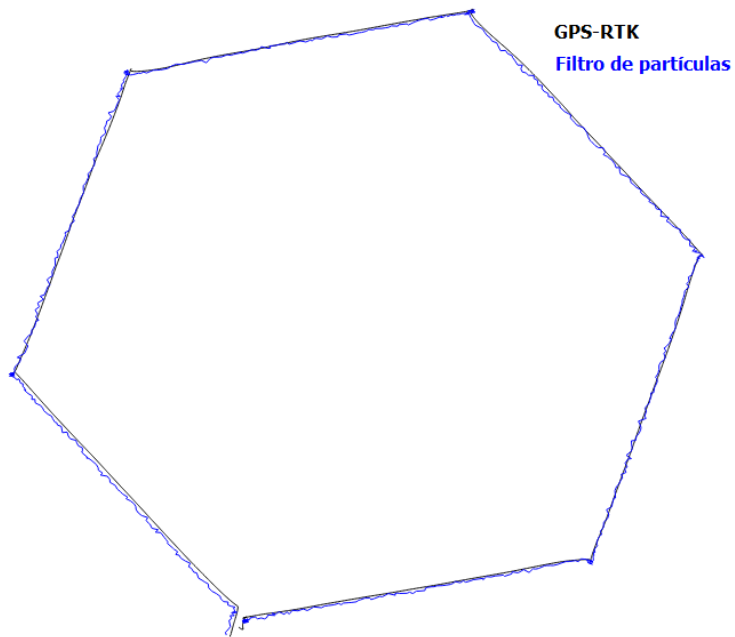


Fonte: Produção do autor.

Uma vez descrito o funcionamento do FP, que foi usado neste experimento, torna-se importante mostrar o resultado de sua aplicação, que consta na Figura 8.7. Na Figura 8.7, a trajetória estimada pelo FP é denotada pela cor azul e a trajetória média pelo GPS-RTK é denotada pela cor preta.

Pela Figura 8.7, percebe-se que a estimação da trajetória pelo FP obteve proximidade semelhante aos resultados mostrados no Capítulo 7. Desta forma, torna-se

Figura 8.7 - Estimação da trajetória por FP.



Fonte: Produção do autor.

importante mostrar os incertezas associadas a esta estimação da trajetória, que são mostradas na Tabela 8.2.

Tabela 8.2 - Incertezas para estimação da trajetória por FP.

Coordenada	Valor da incerteza
Erro médio	
Latitude	34,04cm
Longitude	91,31cm
Desvio	
Latitude	25,74cm
Longitude	54,05cm
Variância	
Latitude	662,85cm
Longitude	2.921,51cm
Erro médio na trajetória	
-	101,86cm

### 8.3 Considerações sobre a quantificação de incertezas

O Capítulo de quantificação de incertezas, na Sessão 8.1, mostrou uma maneira de fornecer a incerteza durante a navegação autônoma por fusão de dados. Sobre este experimento, cabe informar que os conjuntos de desvios utilizados para treinamento, validação e generalização da MLP, foram gerados a partir da trajetória estimada por fusão de dados calculada em CPU. Também é importante ressaltar que esta MLP não foi construída com FPGA.

Ainda sobre o experimento da Sessão 8.1, a forma de separação dos conjuntos de treinamento, validação e generalização, que foram usados no treinamento e teste da MLP, seguiu o mesmo esquema descrito no Capítulo 6.

Com relação ao experimento da Sessão 8.2, cabe ressaltar que o FP é uma técnica consolidada para a estimação de estados em um modelo de espaço de estados não-lineares. Porém, na literatura, é apontado um problema com relação a formulação do algoritmo, em que este problema encontra-se na aplicação da função de verossimilhança na etapa de estimação, ou seja, no operador de verossimilhança. Este problema já foi estudado e a solução proposta foi utilizar a distribuição de Tsallis, como alternativa a outras distribuições, como a Gaussiana, Levi ou Cauchy (CAMPOS VELHO; FURTADO, 2011).

A distribuição Gaussiana é muito usada em problemas de estimação Bayesiana. Justifica-se seu uso devido ao teorema do limite central, na hipótese de que trata-se de um processo estocástico de variáveis aleatórias, estatisticamente independentes e com todos os seus momentos estatísticos bem definidos. Porém, quando se está em uma situação, em que não se tem informações como variância ou outros momentos estatísticos definidos, não é adequado o uso de uma distribuição Gaussiana para calcular a FDP,  $p(x_t|y_t)$ . Por isso, opta-se pela distribuição de Tsallis (CAMPOS VELHO; FURTADO, 2011).

Com a distribuição Tsallis é possível ao algoritmo de FP encontrar a distribuição aproximada de  $p(x_t|y_t)$ , mesmo que esta não possua variância definida. A forma não-extensiva para a entropia de Tsallis é apresentada na Equação 8.2:

$$S_q(p) = \frac{k}{q-1} \left[ 1 - \sum_{i=1}^N (p_i)^q \right] \quad (8.2)$$

onde  $p_i$  é a probabilidade e  $q$  é um parâmetro livre denominado parâmetro de não-

extensividade. No experimento realizado com o FP, utilizou-se  $q = 0,8$ . Na termodinâmica, o parâmetro  $k$  é conhecido como constante de Boltzmann. No limite de  $q \rightarrow 1$ , a entropia de Tsallis se reduz à formula usual de Boltzmann-Gibbs-Shannon.

Entre os parâmetros, que devem ser atribuídos ao algoritmo do FP, destaca-se a quantidade de partículas usadas, que foram 500 partículas no total. Também cabe informar que, na etapa de seleção, foram selecionadas as 97% melhores, ou seja, as que obtiveram um índice de verossimilhança menor ou igual a 3%, não foram usadas na estimação da posição. Os parâmetros aplicados no FP foram obtidos por experimentação numérica.

Sobre o experimento com FP, também é importante informar que as incertezas mostradas na Tabela 8.2 foram calculadas sobre todos os dados, que denotam a trajetória percorrida pelo VANT. Isto é diferente do cálculo feito para as incertezas da fusão de dados, em que se utilizou apenas o conjunto de teste.





## 9 CONCLUSÃO

Os experimentos realizados justificaram a necessidade da fusão de dados para navegação autônoma de VANT. A metodologia proposta provou diminuir as incertezas da estimação da posição e, conseqüentemente, da trajetória estimada. Isto foi, paulatinamente, demonstrado a cada experimento.

Dado a divisão em fases e etapas, cabe concluir de maneira sequencial. Assim, na primeira etapa do Capítulo 6, foi demonstrado que para construção do fusor neural não basta apenas a aplicação dos dados de maneira aleatória no treinamento da MLP. Nesta etapa, foram testados, massivamente, duas abordagens de separação dos dados e foi possível evidenciar que, dependendo da forma como os dados são separados, o desempenho do fusor neural pode ser melhorado ou piorado.

Um destaque na primeira etapa do Capítulo 6, foi a demonstração da importância da inclusão de dados, que contêm possíveis informações sobre o erro de deriva do INS, no conjunto de treinamento aplicado ao MPCA. Incluir estes dados no treinamento alterou positivamente o resultado da estimação feita pela MLP, impactando, inclusive, sobre as incertezas, que denotam intervalo de confiança. Desta forma, conclui-se que a organização dos dados afeta diretamente o treinamento da MLP. Isto porque, cada dado possui sua parcela de contribuição, que influencia diretamente na estimação da atitude da aeronave.

Na segunda etapa, do Capítulo 6, foi verificado o impacto provocado pela presença de ruído. Para abrandar este problema, foram testadas duas abordagens de remoção de ruído, uma baseada na remoção das variações rápidas do INS e a outra na aplicação de filtros lineares.

A abordagem que apresentou maior impacto, ou seja, provocou a melhora mais significativa no treinamento da MLP, foi a que resultou na remoção das variações rápidas do INS. Controlar estas variações é algo difícil, porque elas podem ser provocadas por fatores externos, que afetam o deslocamento do VANT, como, por exemplo, rajadas de vento ou ainda pela própria eletrônica dos sensores. Contudo, os experimentos demonstraram que considerar estas informações, no conjunto de treinamento, piora os resultados da estimação da posição. Portanto, conclui-se que estes dados devem ser considerados como *outliers* e precisam ser removidos.

A etapa de aplicação de filtros lineares testou quatro diferentes tipos de filtros. Nesta etapa, tomou-se o cuidado de aplicar a filtragem com o mínimo de degradação, ou

seja, o mínimo de intensidade nos filtros. Isto porque, durante uma situação de voo, não é possível aplicar filtros baseados em janelas deslizantes, uma vez que não se tem a série temporal completa das medidas do INS para a trajetória, tampouco se tem a estimação completa da trajetória por imagens, o que se tem são informações em instantes de tempo discretos. Portanto, conclui-se que a MLP deve ser treinada com dados mais próximos possíveis da forma como são coletados durante o voo.

Outro ponto é que se manteve o tamanho da janela deslizante dos filtro em igual para todos os dados. Na literatura, é encontrado a aplicação de filtros lineares, variando o tamanho da janela deslizante para cada tipo de dado, o que implica na construção de mais de um modelo de inteligência artificial, para treinar com cada dado filtrado ou pelo menos para cada coordenada filtrada. Nesta Tese, foi gerado apenas um modelo de inteligência artificial, o que garante a redução da complexidade do fusor e auxilia na construção do *hardware* dedicado com FPGA.

O destaque na etapa de aplicação de filtros lineares é que quando foi aplicada a filtragem linear, junto com a remoção das variações rápidas do INS, o resultado da estimação foi piorado significativamente. Isto gerou um aumento de 17% do erro na estimação da trajetória, em relação a etapa de remoção das variações rápidas do INS, o que foi mostrado na Tabela 6.11. Portanto, conclui-se que não é interessante aplicar as duas formas de filtragem conjuntamente.

Por fim, o Capítulo 6 focou em demonstrar que a MLP autoconfigurada pode emular o FK, quando aplicada como fusor de dados na navegação autônoma de VANT. Além disso, dado que o FK usado como referência foi próximo a um GPS de alta precisão, isto estimulou a próxima fase de experimentos com a premissa de que a MLP pode emular o próprio GPS de alta precisão. A Tabela 9.1, resume os melhores resultados das etapas do Capítulo 6.

Tabela 9.1 - Evolução dos resultados no Capítulo 6.

Coordenada	Dados aleatórios	Seleção de dados	Filtragem
Erro médio por coordenada			
Latitude	47,57cm	35,74cm	30,14cm
Longitude	62,83cm	53,73cm	33,27cm
Erro médio na trajetória			
-	87,38cm	69,41cm	49,19cm

Na Tabela 9.1, a coluna denominada dados aleatórios mostra o resultado para a

primeira aplicação do fusor neural, sem a seleção das técnicas de validação cruzada. Já a coluna denominada seleção de dados mostra o melhor resultado com a aplicação das técnicas de validação cruzada. Por último, é mostrado o melhor resultado com as técnicas de filtragem.

Na primeira etapa, do Capítulo 7, foi estudado a aplicação da visão computacional para estimação da posição do VANT durante o voo. Foram testados três extratores de bordas, sendo um deles uma MLP autoconfigurada. Além dos extratores também foi aplicado uma técnica de filtragem de destaque na literatura. Os resultados mostraram que a navegação autônoma por processamento de imagens não foi eficaz, pois apresentou erros muito altos. A conclusão desta etapa é que não é possível ainda confiar apenas na navegação por imagens com correlação cruzada, como solução para navegação autônoma.

Na segunda etapa, do Capítulo 7, ocorreu a inclusão dos sensores inerciais. Foi nesta etapa que a metodologia completa foi aplicada, para o experimento no Brasil. Os resultados da fusão de dados foram significativamente melhores do que a navegação apenas por imagens, o que demonstra a necessidade da fusão de dados para navegação autônoma de VANT. Portanto, conclui-se que a fusão de dados é uma etapa que não pode ser negligenciada, sob pena de inviabilizar a navegação autônoma.

Ainda na segunda etapa do Capítulo 7, foi aplicado o mesmo procedimento de filtragem utilizado no Capítulo 6. Isto porque, apesar dos resultados da fusão de dados estarem melhores do que os resultados da navegação apenas por imagens, os erros ainda estavam altos em comparação com os resultados do Capítulo 6 à medida que faltava a filtragem nos dados inerciais. Após a aplicação da filtragem, percebeu-se uma melhora significativa, em que o erro médio na estimação da trajetória foi de 118cm, sem filtragem este erro era de 555cm.

Como informado anteriormente, o erro de posicionamento do INS para o VANT usado no Brasil é de 200cm. Isto significa que após a filtragem dos dados inerciais, a imprecisão da fusão de dados foi 50% menor que o INS. A partir disto, conclui-se a eficácia da fusão de dados, em particular, a eficácia do fusor neural.

Outra questão que a segunda etapa do Capítulo 7 demonstrou foi a independência da metodologia com relação ao tipo de VANT utilizado. Isto pode ser notado uma vez que os dados usados em cada fase foram coletados por VANTs completamente diferentes. A Tabela 9.2, resume a evolução dos erros médios para as duas primeiras etapas do Capítulo 7.

Tabela 9.2 - Evolução dos resultados no Capítulo 7.

Coordenada	Navegação por imagens	Fusão de dados
Erro médio por coordenada		
Latitude	1.656,67cm	41,29cm
Longitude	3.154,50cm	111,33cm
Erro médio na trajetória		
-	3.617,96	118,89cm

A terceira etapa do Capítulo 7 demonstrou o impacto de implementar o fusor neural em hardware dedicado com FPGA. Foram testados dois dispositivos pequenos e aptos para embarque em VANTs de pequeno porte. Nesta etapa, ficou evidente a necessidade de usar o MPCA, pois o tamanho da arquitetura da MLP definiu se o fusor neural coube ou não na FPGA. Desta forma, conclui-se que para construção de uma MLP com FPGA, em dispositivos pequenos, é necessário aplicar alguma metodologia para encontrar uma arquitetura otimizada ou subótima, que, nesta Tese, foi o MPCA.

Na terceira etapa, do Capítulo 7, também foram testadas três formas de implementar a função de ativação da rede neural. Com isto, ficou evidente o impacto que pode ocorrer nas estimações, principalmente nas incertezas que denotam intervalo de confiança, dependendo da forma que a função de ativação é construída com FPGA. A Tabela 9.3 resume a evolução dos erros médios para cada implementação da função de ativação. Na Tabela 9.3, os erros médios são comparados com a fusão de dados em CPU. A conclusão desta etapa foi a construção de uma função de ativação com LUT que fosse simples e que obtivesse resultado próximo a fusão de dados feita em CPU.

Tabela 9.3 - Evolução dos resultados em *hardware* dedicado.

Coordenada	CPU	LUT	CORDIC	LUT com interpolação
Erro médio por coordenada				
Latitude	41,29cm	190,45cm	55cm	63,96cm
Longitude	111,33cm	215,68cm	103,86cm	47,57cm
Erro médio na trajetória				
-	118,89cm	307,88cm	109,50cm	88,70cm

Na primeira etapa, do Capítulo 8, foi proposto uma forma de estimar a incerteza para fusão de dados durante a navegação autônoma. Isto porque, além de fornecer

a estimação da posição, é importante também entregar ao computador de bordo, algo que indique se a fusão de dados está adequada ou não. Para isto, alterou-se a arquitetura do fusor neural para que fosse possível entregar, além das estimações de cada coordenada, também uma estimação do desvio padrão, associado ao erro médio da estimação da trajetória. Conclui-se que a abordagem proposta conseguiu entregar uma forma de segurança para a aplicação da fusão de dados.

Na segunda etapa, do Capítulo 8, foi aplicado o FP para estimação da posição do VANT. Este filtro foi construído com base no erro médio de medida do INS, para cada coordenada. Pelos resultados do FP, foi possível concluir que a estimação bayesiana a partir do erro médio do INS é tão boa quanto a fusão de dados. Contudo, cabe evidenciar que o INS usado possui uma precisão alta, o que indica que este resultado não necessariamente é definitivo para qualquer VANT. A Tabela 9.4, mostra um resumo dos melhores erros médios para fusão de dados em CPU, FPGA e para a estimação da posição por FP.

Tabela 9.4 - Resultados da estimação da posição por abordagem.

Coordenada	CPU	FPGA	FP
Erro médio por coordenada			
Latitude	41,29cm	63,96cm	34,04cm
Longitude	111,33cm	47,57cm	91,31cm
Erro médio na trajetória			
-	118,89cm	88,70cm	101,86cm

Sumarizando as contribuições da presente tese, destaca-se:

- Fusão de dados baseada em redes artificiais.
- Fusão de dados de inercial e visão computacional baseada em filtro de partículas, o que permite quantificar a incerteza na estimação de posicionamento.
- Identificação de incerteza por redes neurais em fusão dos dados.
- Implementação de redes neurais MLP em 2 arquiteturas para FPGA: coprocessador distinto da CPU e FPGA acoplados no mesmo chip.
- Implementação do algoritmo CORDIC em substituição a estratégia de LUT em FPGA.

Por fim, os resultados obtidos indicam que usar múltiplas fontes de dados traz benefícios a problemática da navegação autônoma de VANT e que a metodologia proposta pode ser considerada mais uma ferramenta testada e, portanto, disponível a comunidade científica.

Esta Tese gerou publicações científicas, oriundas diretamente dos experimentos realizados, são elas: [PENHA NETO et al. \(2018\)](#), [PENHA NETO et al. \(2019\)](#), [PENHA NETO et al. \(2019\)](#), [PENHA NETO et al. \(2019\)](#), [PENHA NETO et al. \(2020\)](#)

## 9.1 Trabalhos futuros

Toda pesquisa científica tende a continuar e, com isso, produzir melhores resultados. Neste sentido, indica-se como trabalhos futuros:

- Testes da metodologia com outras metodologias de localização por imagens;
- Testes da metodologia com outros sensores imageadores, prevendo outras coberturas de solo;
- Na fusão de dados, utilizar a combinação de mais sensores;
- Na estimação da posição por imagens, a implementação do algoritmo de correlação cruzada com técnicas de programação de alto desempenho para melhorar o tempo de processamento do método de estimação de posição. Além disso, para melhorar a estimação da posição fornecida pelo algoritmo, realizar uma pré-classificação das imagens, por exemplo, com o uso da rede Mapa Auto-Organizáveis de Kohonen ([MOTA et al., 2014](#));
- Realizar testes com outras meta-heurísticas para determinar da arquitetura da MLP.

## REFERÊNCIAS BIBLIOGRÁFICAS

- ADUSUMILLI, S.; BHATT, D.; WANG, H.; DEVABHAKTUNI, V.; BHATTACHARYA, P. A novel hybrid approach utilizing principal component regression and random forest regression to bridge the period of gps outages. **Neurocomputing**, v. 166, p. 185–192, 2015. [9](#), [21](#)
- AGÊNCIA NACIONAL DE AVIAÇÃO CIVIL. **Drones**. 2020. Disponível em: <[www.anac.gov.br/assuntos/paginas-tematicas/drones](http://www.anac.gov.br/assuntos/paginas-tematicas/drones)>. Acesso em: 30 dez. 2020. [2](#), [122](#)
- AL-KAFF, A.; MARTÍN, D.; GARCÍA, F.; ESCALERA, A.; ARMINGOL, J. M. Survey of computer vision algorithms and applications for unmanned aerial vehicles. **Expert Systems with Applications**, v. 92, p. 447–463, 2018. [2](#), [7](#), [18](#), [29](#), [30](#), [31](#), [33](#), [34](#), [37](#), [38](#), [39](#), [40](#), [104](#), [111](#)
- ALILAT, F.; YAHIAOUI, R. Mlp on fpga: optimal coding of data and activation function. In: IEEE INTERNATIONAL CONFERENCE ON INTELLIGENT DATA ACQUISITION AND ADVANCED COMPUTING SYSTEMS: TECHNOLOGY AND APPLICATIONS (IDAACS), 10., 2019. **Proceedings...** [S.l.]: IEEE, 2019. p. 525–529. [55](#)
- ANDRAKA, R. A survey of cordic algorithms for fpga based computers. In: INTERNATIONAL SYMPOSIUM ON FIELD PROGRAMMABLE GATE ARRAYS, 2001. **Proceedings...** [S.l.], 2001. [136](#), [137](#)
- ANOCHI, J.; CAMPOS VELHO, H. F.; FURTADO, H.; LUZ, E. P. Self-configuring two types of neural networks by mpca. **Journal of Mechanics Engineering and Automation**, v. 5, p. 112–120, 2015. [9](#), [55](#), [56](#), [70](#), [74](#), [79](#)
- ANTONY, R. T. **Data fusion support to activity-based intelligence**. [S.l.]: Artech House, 2015. [45](#), [46](#)
- ARTHUR, M. P. Detecting signal spoofing and jamming attacks in uav networks using a lightweight ids. In: INTERNATIONAL CONFERENCE ON COMPUTER, INFORMATION AND TELECOMMUNICATION SYSTEMS (CITS), 2019. **Proceedings...** [S.l.], 2019. [3](#), [29](#)
- BAENA, S.; BOYD, D.; MOAT, J. Uavs in pursuit of plant conservation - real world experiences. **Ecological Informatics**, v. 47, p. 2–9, 2017. [16](#)

BALAN, N.; LIU, L. B.; LE, H. J. A brief review of equatorial ionization anomaly and ionospheric irregularities. **Earth and Planetary Physics**, v. 2, p. 4, 2018. **5, 103**

BAO, S.; LAI, J.; CHEN, Z.; LYU, P.; CHEN, W. Aerodynamic model/ins/gps failure-tolerant navigation method for multirotor uavs based on federated kalman filter. In: CHINESE AUTOMATION CONGRESS, 2017. **Proceedings...** [S.l.], 2017. **21, 29, 47, 99**

BARROS, D.; TAKAHASHI, H.; WRASSE, M. W.; FIGUEIREDO, C. A. O. B. Characteristics of equatorial plasma bubbles observed by tec map based on ground-based gnss receivers over South America. **Annales Geophysicae**, v. 36, p. 91–100, 2018. **103**

BLOM, D. J. **Unmanned aerial systems: a historical perspective**. [S.l.]: Combat Studies Institute Press, 2010. **15**

BRAGA, J. R. G. **Navegação autônoma de vant por imagens lidar**. Tese (Doutorado em Computação e Matemática Aplicada — Instituto Nacional de Pesquisas Espaciais, São José dos Campos - SP, 2018. **6, 7, 18, 19, 24, 29, 34, 37, 38, 39, 42, 43, 58, 69, 91, 103, 104, 107, 108, 111, 112, 113, 122, 123, 124, 125, 126, 127, 131, 132, 150, 151, 153**

BRAGA, J. R. G.; CAMPOS VELHO, H. F.; CONTE, G.; DOHERTY, P.; SHIGUEMORI, E. H. An image matching system for autonomous uav navigation based on neural network. In: INTERNATIONAL CONFERENCE ON CONTROL, AUTOMATION, ROBOTICS AND VISION, 2016. **Proceedings...** [S.l.], 2016. **33, 39, 62, 69, 91, 103, 107, 108, 111**

BRAGA, J. R. G.; CAMPOS VELHO, H. F.; SHIGUEMORI, E. H. Estimation of uav position using lidar images for autonomous navigation over the ocean. In: ICST INTERNATIONAL CONFERENCE ON SENSING TECHNOLOGY, 2015. **Proceedings...** [S.l.], 2015. **33, 91**

BRAGA, J. R. G.; CAMPOS VELHO, H. F.; SHIGUEMORI, E. H.; DOHERTY, P. Drone autonomous navigation by hardware image processing. In: ARGENTINE CONGRESS ON COMPUTATIONAL MECHANICS, 2019. **Proceedings...** [S.l.], 2019. **19, 30, 32, 33, 34, 43, 56, 62**

BRAGA, J. R. G.; SHIGUEMORI, E. H.; CAMPOS VELHO, H. F. Determining the trajectory of unmanned aerial vehicles by a novel approach for the particle filter. **Mecânica Computacional**, v. 15, 2018. **62, 104**



CAMPOS VELHO, H. F.; FURTADO, H. Adaptive particle filter for stable distribution. In: **CONSTANDA, C. AND HARRIS, P. J. INTEGRAL METHODS IN SCIENCE AND ENGINEERING: COMPUTATIONAL AND ANALYTIC ASPECTS**. Boston: Birkhä, 2011. [153](#), [154](#), [155](#), [158](#)

CARVALHO, A.; RAMOS, F.; CHAVES, A. Metaheuristics for the feedforward artificial neural network (ann) architecture optimization problem. **World Congress of Structural and Multidisciplinary Optimization**, v. 20, p. 1273–1284, 2011. [56](#)

CHEN, X.; XUELONG, Z.; WANG, Z.; LI, M.; OU, Y.; YUFAN, S. Multi-frequency data fusion for attitude estimation based on multi-layer perception and cubature kalman filter. **IEEE Access**, v. 8, p. 144373–144381, 2020. [24](#), [50](#)

CHIROLI, D. G.; MONTILHA, M.; SAMED, M. Feasibility analysis of uav in dengue control. **Independent Journal of Management & Production**, v. 8, n. 2, 2017. [16](#)

CILLIERS, P.; OPPERMAN, B.; MEYER, R. Investigation of ionospheric scintillation over south africa and the south atlantic anomaly using gps signals: first results. In: IEEE INTERNATIONAL GEOSCIENCE AND REMOTE SENSING SYMPOSIUM, 2009. **Proceedings...** [S.l.], 2009. [5](#), [6](#)

CINTRA, R.; CAMPOS VELHO, H. F. Data assimilation by neural network under missing data. In: CONFERENCE OF COMPUTATIONAL INTERDISCIPLINARY SCIENCE, 30., 2019. **Proceedings...** [S.l.], 2019. p. 379–388. [52](#)

CONTE, G.; DOHERTY, P. An integrated uav navigation system based on aerial image matching. In: IEEE AEROSPACE CONFERENCE, 2008. **Proceedings...** [S.l.], 2008. [32](#), [63](#), [69](#), [83](#), [84](#), [91](#), [99](#), [103](#), [107](#), [108](#)

\_\_\_\_\_. Vision-based unmanned aerial vehicle navigation using geo-referenced information. **Journal on Advances in Signal Processing**, 2009. [8](#), [10](#), [20](#), [24](#), [31](#), [33](#), [57](#), [59](#), [63](#), [64](#), [65](#), [66](#), [69](#), [83](#), [84](#), [91](#), [99](#), [103](#), [104](#), [123](#)

CORKE, P. **Robotics, vision and control: fundamental algorithms in matlab**. [S.l.]: Springer, 2011. [1](#)

CORRADO, C.; PANETTA, K. Data fusion and unmanned aerial vehicles (uavs) for first responders. In: IEEE INTERNATIONAL SYMPOSIUM ON

TECHNOLOGIES FOR HOMELAND SECURITY (HST), 2017. **Proceedings...** [S.l.], 2017. 6, 9, 20

CRISTIANINI, N. **Cross-validation (k-fold cross-validation, leave-one-out, jackknife, bootstrap)**. [S.l.: s.n.], 2004. 75

DARYANAVARD, H.; HARIFI, A. Implementing face detection system on uav using raspberry pi platform. In: IRANIAN CONFERENCE ON ELECTRICAL ENGINEERING, 2018. **Proceedings...** [S.l.], 2018. 2, 17, 124

DASGUPTA, S. **Neural networks using c#: from basic perceptrons to fully functional feedforward multilayer perceptrons**. [S.l.]: eBook Kindle, 2020. 51

DASH, J. P.; WATT, M. S.; PEARSE, G. D.; HEAPHY, M.; DUNGEY, H. S. Assessing very high resolution uav imagery for monitoring forest health during a simulated disease outbreak. **ISPRS Journal of Photogrammetry and Remote Sensing**, v. 131, p. 1–14, 2017. 2, 16

DATTA-BARUA, S.; DOHERTY, P.; DEHEL, T.; KLOBUCHAR, J. Ionospheric scintillation effects on single and dual frequency gps positioning. In: ION GPS 2003, 2003. **Proceedings...** Porland, OR., 2003. 63

DIGILENT. **Start**. 2020. Disponível em: <[www.reference.digilentinc.com/reference/programmable-logic/zybo-z7/start](http://www.reference.digilentinc.com/reference/programmable-logic/zybo-z7/start)>. Acesso em: 30 dez. 2020. 131, 132

DJI. **Matrice600 pro**. 2020. Disponível em: <[www.dji.com/br/matrice600-pro](http://www.dji.com/br/matrice600-pro)>. Acesso em: 30 dez. 2020. 65, 66, 104, 115

\_\_\_\_\_. **Phantom 4 rtk**. 2020. Disponível em: <[www.dji.com/br/phantom-4-rtk](http://www.dji.com/br/phantom-4-rtk)>. Acesso em: 30 dez. 2020. 105

DOMICIAN, M. A. P. **Navegação aérea autônoma baseada em modelos digitais de elevação**. Tese (Doutorado em Engenharia Eletrônica e Computação) — Instituto Tecnológico de Aeronáutica, São José dos Campos - SP, 2018. 2, 6, 104

ELECTRICAL, I. O.; IEEE, E. E. **Ieee draft standard for transparency of autonomous systems**. [S.l.]: IEEE, 2020. 1, 2, 7

ELEZI, E.; CANKAYA, G.; BOYACI, A.; YARKAN, S. The effect of electronic jammers on gps signals. In: 16TH INTERNATIONAL MULTI-CONFERENCE

ON SYSTEMS, SIGNALS DEVICES (SSD), 2019. **Proceedings...** [S.l.], 2019. 3, 29

ENAMAMU, T. S.; CLARKE, N.; HASKELL-DOWLAND, P.; LI, F. Smart watch based body-temperature authentication. In: INTERNATIONAL CONFERENCE ON COMPUTING NETWORKING AND INFORMATICS (ICCNI), 2017. **Proceedings...** [S.l.], 2017. 45

FARIA, L.; SILVESTRE, C.; CORREIA, M. Gps-dependent systems: vulnerabilities to electromagnetic attacks. **Journal of Aerospace Technology and Management**, v. 8, n. 4, p. 423–430, 2016. 29

FILYASHKIN, N. K.; YATSKIVSKY, V. S. Prediction of inertial navigation system error dynamics in ins/gps system. In: IEEE INTERNATIONAL CONFERENCE ACTUAL PROBLEMS OF UNMANNED AIR VEHICLES DEVELOPMENTS PROCEEDINGS, 2., 2013. **Proceedings...** [S.l.], 2013. 82

FIORI, L.; DOSHI, A.; MARTINEZ, E.; ORAMS, M. B.; BOLLARD-BREEN, B. The use of unmanned aerial systems in marine mammal research. **Remote Sensing**, v. 9, n. 6, 2017. 2, 17

FOLEY, J. D.; DAM, V. A.; FEINER, K. S. **Computer graphics: principles and practice**. [S.l.]: Addison-Wesley Professional, 2013. 34, 35, 36, 37, 38, 39

FORNARI, G.; SANTIAGO JÚNIOR, V.; SHIGUEMORI, E. H. A self-adaptive approach for autonomous uav navigation via computer vision. In: COMPUTATIONAL SCIENCE AND ITS APPLICATIONS, 2018. **Proceedings...** [S.l.], 2018. 20

FOURATI, H. **Multisensor data fusion: from algorithms and architectural design to applications**. [S.l.]: CRC Press, 2020. 45, 46, 47

FRADEN, J. **Handbook of modern sensors**. [S.l.]: Springer-Verlag New York, 2010. 3, 45, 46, 69, 84, 91, 97, 98

FUHRMANN, M.; HOROWITZ, M. C. Droning on: explaining the proliferation of unmanned aerial vehicles. **International Organization**, 2017. 2, 98

FURTADO, H. C. M. **Redes neurais e diferentes métodos de assimilação de dados em dinâmica não linear**. Dissertação (Mestrado em Computação Aplicada) — Instituto Nacional de Pesquisas Espaciais, São José dos Campos - SP, 2008. 150, 151, 156

GAIKWAD, N. B.; TIWARI, V.; KESKAR, A.; SHIVAPRAKASH, N. C. Efficient fpga implementation of multilayer perceptron for real-time human activity classification. **IEEE Access**, v. 7, p. 26696–26706, 2019. 52, 54

GANDHI, G. M.; SALVI. Artificial intelligence integrated blockchain for training autonomous cars. In: INTERNATIONAL CONFERENCE ON SCIENCE TECHNOLOGY ENGINEERING AND MATHEMATICS, 2019. **Proceedings...** [S.l.], 2019. 2

GAO, X.; JIA, H.; CHEN, Z.; YUAN, G.; YANG, S. Uav security situation awareness method based on semantic analysis. In: IEEE INTERNATIONAL CONFERENCE ON POWER, INTELLIGENT COMPUTING AND SYSTEMS, 2020. **Proceedings...** [S.l.], 2020. 2, 3, 17

GEDRAITE, S. E.; HADAD, M. Investigation on the effect of a gaussian blur in image filtering and segmentation. In: PROCEEDINGS ELMAR-2011, 2011. **Proceedings...** [S.l.], 2011. 37, 108

GEVAERT, C. M.; TANG, J.; GARCÍA-HARO, F. J.; SUOMALAINEN, J.; KOOISTRA, L. Combining hyperspectral uav and multispectral formosat-2 imagery for precision agriculture applications. In: WORKSHOP ON HYPERSPECTRAL IMAGE AND SIGNAL PROCESSING: EVOLUTION IN REMOTE SENSING, 6., 2014. **Proceedings...** [S.l.], 2014. 2, 16

GREWAL, M.; ANDREWS, A.; BARTONE, C. **Global navigation satellite systems, inertial navigation, and integration**. [S.l.]: Wiley, 2019. 3, 4

GROVES, P. D. Principles of gnss, inertial, and multisensor integrated navigation systems. **IEEE Aerospace and Electronic Systems Magazine**, 2015. 3, 4, 20, 25

GUAN, X.; LV, R.; LI, R.; WANG, M.; LIU, H. Verification platform for formation method based on quad-rotor uav. In: CHINESE CONTROL AND DECISION CONFERENCE, 2020. **Proceedings...** [S.l.], 2020. 124

GUANG, H.; JI, L. Bayesian state estimation in sensorimotor systems with particle filtering. **IEEE Transactions on Neural Systems and Rehabilitation Engineering**, v. 28, n. 7, p. 1528–1538, 2020. 10, 150, 151, 153

GULATI, I.; LI, H.; STANTON, S.; JOHNSTON, M.; DLAY, S. Investigation of ionospheric phase scintillation at middle-latitude receiver station. In:

INTERNATIONAL SYMPOSIUM ELMAR, 2019. **Proceedings...** [S.l.], 2019. 5, 6

HAYKIN, S. S. **Neural networks and learning machines.** [S.l.]: Pearson Education, 2009. 9, 22, 23, 42, 52

HISHAM, M. B.; YAAKOB, S. N.; RAO, R. A. A.; NAZREN, A. B. A.; WAFI, N. M. Template matching using sum of squared difference and normalized cross correlation. In: IEEE STUDENT CONFERENCE ON RESEARCH AND DEVELOPMENT (SCORED), 2015. **Proceedings...** [S.l.], 2015. 32

HOFMANN-WELLENHOF, B.; LICHTENEGGER, H.; WASLE, E. **Gnss: global navigation satellite systems: gps, glonass, galileo, and more.** [S.l.: s.n.], 2007. 25

HUANG, Y.; ZHANG, Y.; XIAO, H. Multi-robot system task allocation mechanism for smart factory. In: JOINT INTERNATIONAL INFORMATION TECHNOLOGY AND ARTIFICIAL INTELLIGENCE CONFERENCE, 8., 2019. **Proceedings...** [S.l.], 2019. 1

HUTTUNEN, M. Unmanned, remotely piloted, or something else? analysing the terminological dogfight. **Air and Space Law**, v. 42, n. 3, p. 349–368, 2017. 2, 122

JAIN, A.; GUPTA, R. Gaussian filter threshold modulation for filtering flat and texture area of an image. In: INTERNATIONAL CONFERENCE ON ADVANCES IN COMPUTER ENGINEERING AND APPLICATIONS, 2015. **Proceedings...** [S.l.], 2015. 37, 40, 108

JAIN, R.; KASTURI, R.; SCHUNCK, B. G. **Machine vision.** [S.l.]: McGraw-Hill, 1995. 40, 41, 42

JIN, W. C.; KIM, K.; CHOI, W. J. Robust jamming algorithm for location-based uav jamming system. In: IEEE ASIA-PACIFIC MICROWAVE CONFERENCE (APMC), 2019. **Proceedings...** [S.l.], 2019. 3, 29

JULIO, O. R.; SOARES, L. B.; COSTA, E. A. C.; BAMPI, S. Energy-efficient gaussian filter for image processing using approximate adder circuits. In: IEEE INTERNATIONAL CONFERENCE ON ELECTRONICS, CIRCUITS, AND SYSTEMS, 2015. **Proceedings...** [S.l.], 2015. 37, 108

JUMAAT, A. K.; YASIRAN, S. S.; MALEK, A. A.; RAHMAN, W. E. Z. W. A.; BADRIN, N.; OSMAN, S. H.; RAFIEE, S. R.; MAHMUD, R. Performance

comparison of canny and sobel edge detectors on balloon snake in segmenting masses. In: INTERNATIONAL CONFERENCE ON COMPUTER AND INFORMATION SCIENCES, 2014. **Proceedings...** [S.l.], 2014. 42

KALMAN, R. E. A new approach to linear filtering and prediction problems. **Transactions of the ASME - Journal of Basic Engineering**, n. 8, p. 35–45, 1960. 47, 48

KALRA, A.; CHHOKAR, R. L. A hybrid approach using sobel and canny operator for digital image edge detection. In: INTERNATIONAL CONFERENCE ON MICRO-ELECTRONICS AND TELECOMMUNICATION ENGINEERING, 2016. **Proceedings...** [S.l.], 2016. 39, 42

KEANE, J.; CARR, S. A brief history of early unmanned aircraft. **Johns Hopkins Apl Technical Digest**, p. 559–570, 2013. 15

KHALDI, A.; BENSALAH, C.; BRAHAM, A. C.; CHERKI, B. Uav attitude estimation using visual and inertial data fusion based on observer in  $so(3)$ . In: INTERNATIONAL CONFERENCE ON ADVANCED ELECTRICAL ENGINEERING, 2019. **Proceedings...** [S.l.], 2019. 6, 7, 21

KHALEGHI, B.; KHAMIS, A.; KARRAY, F. O.; RAZAVI, S. N. Multisensor data fusion: a review of the state-of-the-art. **Information Fusion**, v. 14, n. 1, 2013. 7, 20, 44, 45, 59, 69, 70, 84, 91, 150

KIM, D. J.; KIM, M. K.; LEE, K. S.; PARK, H. G.; LEE, M. H. Localization system of autonomous vehicle via kalman filtering. In: INTERNATIONAL CONFERENCE ON CONTROL, AUTOMATION AND SYSTEMS, 11., 2011. **Proceedings...** [S.l.], 2011. 83, 91

KIM, T.; SIN, C. S.; LEE, S. Analysis of effect of spoofing signal in gps receiver. In: INTERNATIONAL CONFERENCE ON CONTROL, AUTOMATION AND SYSTEMS, 2012. **Proceedings...** [S.l.], 2012. 3, 29

KIM, T.; SIN, C. S.; LEE, S.; KIM, J. H. Analysis of effect of anti-spoofing signal for mitigating to spoofing in gps l1 signal. In: INTERNATIONAL CONFERENCE ON CONTROL, AUTOMATION AND SYSTEMS, 2013. **Proceedings...** [S.l.], 2013. 3, 29

KISELEV, E. Methods of training multi-layer perceptron. **Actual directions of scientific researches of the XXI century: theory and practice**, 2015. 60

KONG, X.; FANG, G.; LIU, L.; TRAN, T. P. Low computational data fusion approach using ins and uwb for uav navigation tasks in gps-denied environments. In: INTERNATIONAL CONFERENCE ON PARALLEL AND DISTRIBUTED COMPUTING, APPLICATIONS AND TECHNOLOGIES (PDCAT), 2019. **Proceedings...** [S.l.], 2019. 8, 21, 50

KUTTY, S. B.; SAAIDIN, S.; YUNUS, P. N. A. M.; HASSAN, S. A. Evaluation of canny and sobel operator for logo edge detection. In: INTERNATIONAL SYMPOSIUM ON TECHNOLOGY MANAGEMENT AND EMERGING TECHNOLOGIES, 2014. **Proceedings...** [S.l.], 2014. 42

LAWAL, A. B. **Fundamentals of satellite navigation systems: how to design gps/gnss.** [S.l.]: Independently Published, 2019. 26, 27

LAYMAN, R. D. **Naval aviation in the first world war: its impact and influence.** [S.l.]: Naval Institute Press, 1996. 15

LEE, E. M.; WEE, I.; KIM, T.; SHIM, D. H. Comparison of visual inertial odometry using flightgoggles simulator for uav. In: INTERNATIONAL CONFERENCE ON CONTROL, AUTOMATION AND SYSTEMS, 2019. **Proceedings...** [S.l.], 2019. 19, 109

LEE, K.; LEE, J.; WOO, B.; LEE, J.; LEE, Y.; RA, S. Modeling and control of a articulated robot arm with embedded joint actuators. In: INTERNATIONAL CONFERENCE ON INFORMATION AND COMMUNICATION TECHNOLOGY ROBOTICS, 2018. **Proceedings...** [S.l.], 2018. 1

LI, Q.; LI, R.; JI, K.; DAI, W. Kalman filter and its application. In: INTERNATIONAL CONFERENCE ON INTELLIGENT NETWORKS AND INTELLIGENT SYSTEMS (ICINIS), 2015. **Proceedings...** [S.l.], 2015. p. 74–77. 47, 48

LIGGINS, M.; HALL, D.; LLINAS, J. J. **Handbook of multisensor data fusion: theory and practice.** [S.l.]: CRC Press, 2017. 44, 46, 47

LIMA, S. A. **Avaliação da acurácia do posicionamento e orientação de aeronaves remotamente pilotadas com uso de técnicas de fotogrametria e processamento digital de imagens.** Dissertação (Mestrado em Sensoriamento Remoto) — Instituto nacional de Pesquisas Espaciais, São José dos Campos - SP, 2018. 2, 19, 25, 26, 105, 106, 122



LIU, F.; ZHANG, Q. Application of extended kalman filter to unmanned helicopter navigation. In: WORLD CONGRESS ON INTELLIGENT CONTROL AND AUTOMATION, 2012. **Proceedings...** [S.l.], 2012. 99

MAES, W.; STEPPE, K. Perspectives for remote sensing with unmanned aerial vehicles in precision agriculture. **Trends in Plant Science**, v. 21, n. 2, p. 152–164, 2018. 16

MANOJ, S. **Image registration techniques using mutual information**. [S.l.]: Lambert Academic Publishing, 2020. 32, 36

MARTINS, W. M.; DANTAS, A.; JESUS, L.; SOUZA, A.; RAMOS, A. C. B.; PIMENTA, T. Tracking for inspection in energy transmission power lines using unmanned aerial vehicles (uav): a systematic review of current and specific literature. **IAES International Journal of Robotics and Automation (IJRA)**, v. 9, p. 233–243, 2020. 17

MEI, H.; CHANGYI, H.; GUANGMING, W. Novel method to evaluate the effectiveness of uav navigation spoofing. In: IEEE INTERNATIONAL CONFERENCE ON ELECTRONIC MEASUREMENT INSTRUMENTS (ICEMI), 2019. **Proceedings...** [S.l.], 2019. 3, 29

MELLO, R. F. D.; PONTI, M. A. **Machine learning: a practical approach on the statistical learning theory**. [S.l.]: Springer, 2018. 51, 52

MIRJALILI, S.; FARIS, H. H.; ALJARAH, I. **Evolutionary machine learning techniques: algorithms and applications**. [S.l.]: Springer, 2020. 51

MOHAMMAD, R. **Commercial space technologies and applications: communication, remote sensing, gps, and meteorological satellites**. [S.l.]: CRC Press, 2020. 25, 26, 27, 28, 29

MOHD, T. K.; CARVALHO, J.; JAVAID, A. Y. Multi-modal data fusion of voice and emg data for robotic control. In: IEEE ANNUAL UBIQUITOUS COMPUTING, ELECTRONICS AND MOBILE COMMUNICATION CONFERENCE, 2017. **Proceedings...** [S.l.], 2017. 150

MORSHEDIZADEH, M.; KORDESTANI, M.; CARRIVEAU, R.; TING, D. S.; SAIF, M. Power production prediction of wind turbines using a fusion of mlp and anfis networks. **IET Renewable Power Generation**, v. 12, n. 9, p. 1025–1033, 2018. 24



MOSBAH, H.; EL-HAWARY, M. E. Optimized neural network parameters using stochastic fractal technique to compensate kalman filter for power system-tracking-state estimation. **Transactions on Neural Networks and Learning Systems**, v. 30, p. 379–388, 2019. 50

MOTA, R. L. M.; SHIGUEMORI, E. H.; RAMOS, A. C. B. Application of self-organizing maps at change detection in amazon forest. In: INTERNATIONAL CONFERENCE ON INFORMATION TECHNOLOGY: NEW GENERATIONS, 2014. **Proceedings...** [S.l.], 2014. p. 371–376. 166

NASHMAN, M. The use of vision and touch sensors for dimensional inspection tasks. **The American Society of Mechanical Engineers**, v. 6, n. 2, 1993. 46

NASUDDIN, K. A.; ABDULLAH, M.; HAMID, N. S. A.; HASBI, A. M. Preliminary study on the near equatorial magnetic field model and south atlantic anomaly. In: INTERNATIONAL CONFERENCE ON SPACE SCIENCE AND COMMUNICATION (ICONSPACE), 2015. **Proceedings...** [S.l.], 2015. 5, 6

NEUSYPIN, K. A.; SELEZNEVA, M. S.; PROLETARSKY, A. V.; SHEN, K. Algorithm for building models of ins/gnss integrated navigation system using the degree of identifiability. In: SAINT PETERSBURG INTERNATIONAL CONFERENCE ON INTEGRATED NAVIGATION SYSTEMS, 2018. **Proceedings...** [S.l.], 2018. 82

NGO, T. B.; LE, H. L.; NGUYEN, T. H. Survey of kalman filters and their application in signal processing. In: INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND COMPUTATIONAL INTELLIGENCE, 2009. **Proceedings...** [S.l.], 2009. p. 335–339. 47

NIAN, Z.; JUNG, C. Cnn-based multi-focus image fusion with light field data. In: IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING (ICIP), 2019. **Proceedings...** [S.l.], 2019. 23

NIKUS, K. **Introdução à robótica - análise, controle, aplicações**. [S.l.]: LTC, 2013. 1

NOWOSAD, A. G. **Novas abordagens em assimilação de dados meteorológicos**. Tese (Doutorado em Computação Aplicada) — Instituto Nacional de Pesquisas Espaciais, São José dos Campos - SP, 2001. 9, 52

NURHAYATI; SOEKARNO, I.; HADIHARDAJA, I. K.; CAHYONO, M. A study of hold-out and k-fold cross validation for accuracy of groundwater modeling in

tidal lowland reclamation using extreme learning machine. In: INTERNATIONAL CONFERENCE ON TECHNOLOGY, INFORMATICS, MANAGEMENT, ENGINEERING ENVIRONMENT, 2014. **Proceedings...** [S.l.], 2014. 60, 75

OPANASIUK, Y. A. Inertial measurement unit and external influence duration. In: INTERNATIONAL CONFERENCE ON METHODS AND SYSTEMS OF NAVIGATION AND MOTION CONTROL, 2018. **Proceedings...** [S.l.], 2018. 7, 28, 82

OTSU, N. A threshold selection method from gray-level histograms. **IEEE Transactions on Systems, Man, and Cybernetics**, v. 9, n. 1, p. 62–66, 1979. 39

P., D. A.; KHANDEKAR, M. A. Gps and imu integration on an autonomous vehicle using kalman filter (labview tool). In: INTERNATIONAL CONFERENCE ON INTELLIGENT COMPUTING AND CONTROL SYSTEMS, 2019. **Proceedings...** [S.l.], 2019. 20, 47, 83

PATINO-ESCARCINA, R. E.; COSTA, J. A. F. An evaluation of mlp neural network efficiency for image filtering. In: SEVENTH INTERNATIONAL CONFERENCE ON INTELLIGENT SYSTEMS DESIGN AND APPLICATIONS (ISDA 2007), 2007. **Proceedings...** [S.l.], 2007. p. 335–340. 42

PAULA, E. R.; KHERANI, E.; ABDU, M. A.; BATISTA, I. S.; SOBRAL, J. H.; KANTOR, I. J.; TAKAHASHI, H.; REZENDE, L. F. C.; MUELLA, M.; RODRIGUES, F. S.; KINTNER, P. M.; LEDVINA, B. M.; MITCHELL, C. N.; GROVES, K. Characteristics of the ionospheric irregularities over brazilian longitudinal sector. **Indian Journal of Radio and Space Physics**, 2007. 103

PAULINO, A. C. **Fusão de dados baseada em inteligência computacional híbrida adaptativa aplicada à navegação autônoma de arps em tempo real**. Dissertação (Mestrado em Ciências e Tecnologias Espaciais) — Instituto Tecnológico de Aeronáutica, São José dos Campos - SP, 2019. 6, 7, 8, 9, 22, 59, 65, 66, 69, 71, 83, 84, 91, 99, 104, 107, 122, 150

PEDRINI, H.; SCHWARTZ, W. R. W. R. **Análise de imagens digitais: princípios, algoritmos e aplicações**. [S.l.]: Cengage Learning, 2007. 35, 36, 37, 39, 41, 42, 108

PEDRONI, V. A. **Eletrônica digital moderna e vhdl**. [S.l.]: Campus, 2008. 9, 22, 126, 127, 130, 131, 132

PENHA NETO, G.; CAMPOS VELHO, H. F.; SHIGUEMORI, E. H. Uav autonomous navigation by data fusion and fpga. **Mecânica Computacional**, v. 37, p. 609–618, 2019. 166

\_\_\_\_\_. Uav autonomous navigation by image processing with uncertainty trajectory estimation. In: INTERNATIONAL SYMPOSIUM ON UNCERTAINTY QUANTIFICATION AND STOCHASTIC MODELLING - UNCERTAINTIES 2020, 2020. **Proceedings...** [S.l.]: Springer Nature, 2020. v. 1, p. 211–221. 166

PENHA NETO, G.; PAULINO, A. C.; CAMPOS VELHO, H. F.; OES, L. N. F. G.; SHIGUEMORI, E. H. Computational intelligence for uav autonomous navigation with multisensor data fusion. In: PROCEEDINGS OF THE CONFERENCE OF COMPUTATIONAL INTERDISCIPLINARY SCIENCE (CCIS 2019), 2019. **Proceedings...** [S.l.], 2019. 166

PENHA NETO, G.; SHIGUEMORI, E. H.; CAMPOS VELHO, H. F.; BRAGA., J. R. G. Image processing for uav autonomous navigation applying self-configuring neural network. In: INTERNATIONAL CONFERENCE ON INTEGRAL METHODS IN SCIENCE AND ENGINEERING (IMSE), 2018. **Proceedings...** [S.l.], 2018. 166

\_\_\_\_\_. Autonomous navigation by data fusion with optimized architecture artificial neural networks. In: IBERO-LATIN-AMERICAN CONGRESS ON COMPUTATIONAL METHODS IN ENGINEERING (CILAMCE 2019), 2019. **Proceedings...** [S.l.], 2019. v. 1, p. 1–10. 166

PETUKHOV, N.; ZAMOLODCHIKOV, V.; ZAKHAROVA, E.; SHAMINA, A. Synthesis and comparative analysis of characteristics of complex kalman filter and particle filter in two-dimensional local navigation system. In: SYMPOSIUM ON BIOMEDICAL ENGINEERING, RADIOELECTRONICS AND INFORMATION TECHNOLOGY, 2019. **Proceedings...** [S.l.], 2019. 10, 47, 150, 151, 153

PIMENTA, T. C. **Circuitos digitais: análise e síntese lógica: aplicações em fpga**. [S.l.]: GEN LTC, 2017. 9, 22, 53, 54, 126, 127, 130, 131, 132

PIX4D. **Pix4d**. 2020. Disponível em: <[www.pix4d.com](http://www.pix4d.com)>. Acesso em: 30 dez. 2020. 105

PUTTINAOVARAT, S.; HORKAEW, P. Flood forecasting system based on integrated big and crowdsourced data by using machine learning techniques. **IEEE Access**, 2020. 24

RABINOWICZ, A.; ROSSET, S. Cross-validation for correlated data. **Journal of the American Statistical Association**, 2020. 59, 60, 61, 75

RAO, V. P. S.; RAO, S. G. Design and modelling of anaffordable uav based pesticide sprayer in agriculture applications. **Fifth International Conference on Electrical Energy Systems**, 2019. 2, 16

RASPBERRY PI FOUNDATION. **Raspberrypi**. 2020. Disponível em: <[www.raspberrypi.org](http://www.raspberrypi.org)>. Acesso em: 30 dez. 2020. 123, 124, 125

REBOUÇAS, R. A. **Sistema de posicionamento de vant baseado em imagens**. Dissertação (Mestrado em Computação e Matemática Aplicada) — Instituto Nacional de Pesquisas Espaciais, São José dos Campos - SP, 2019. 6, 104, 109

REVELO, A.; ÁLVAREZ, R.; GRIJALVA, F. Human drowsiness detection in real time, using computer vision. In: FOURTH ECUADOR TECHNICAL CHAPTERS MEETING (ETCM), 2019. **Proceedings...** [S.l.], 2019. p. 1–6. 43

RIZK, M.; MROUE, A.; FARRAN, M.; CHARARA, J. Real-time slam based on image stitching for autonomous navigation of uavs in gnss-denied regions. In: INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE CIRCUITS AND SYSTEMS (AICAS), 2020. **Proceedings...** [S.l.], 2020. p. 301–304. 31

ROOS, D. R. **Aprendizado de máquina aplicado à odometria visual para estimação de posição de veículos aéreos não tripulados**. Dissertação (Mestrado em Ciência da Computação) — Universidade Federal de São Paulo, São José dos Campos - SP, 2018. 30, 104, 109

SACCO, W. F.; DE OLIVEIRA, C. R. E. A new stochastic optimization algorithm based on a particle collision metaheuristic. In: WORLD CONGRESS OF STRUCTURAL AND MULTIDISCIPLINARY OPTIMIZATION, 2005. **Proceedings...** [S.l.], 2005. 55, 56

SAMBATTI, S.; ANOCHI, J.; LUZ, E. P.; SHIGUEMORI, E. H.; CARVALHO, A.; CAMPOS VELHO, H. F. Automatic configuration for neural network applied to atmospheric temperature profile identification. In: XII INTERNATIONAL CONFERENCE ON INTEGRAL METHODS IN SCIENCE AND ENGINEERING, 2012. **Proceedings...** [S.l.], 2012. 50

SAMBATTI, S. B. M.; CAMPOS VELHO, H. F.; FURTADO, H. C. M.; AO, A. S. C. Self-configured neural network for data assimilation using fpga for ocean circulation. **4th Conference of Computational Interdisciplinary Science**, 2016. 22, 24, 126

SANAULLAH, A.; YANG, C.; ALEXEEV, Y.; YOSHII, K.; HERBORDT, M. C. Application aware tuning of reconfigurable multi-layer perceptron architectures. In: IEEE HIGH PERFORMANCE EXTREME COMPUTING CONFERENCE (HPEC), 2018. **Proceedings...** [S.l.], 2018. p. 1–9. 54

SANDINO, J.; PEGG, G.; GONZALEZ, L.; SMITH, G. Aerial mapping of forests affected by pathogens using uavs, hyperspectral sensors, and artificial intelligence. **Sensors**, 2018. 16

SARAIVA, A. G.; SAOTOME, O.; D'AMORE, R.; SHIGUEMORI, E. H. A real time adaptive template matching algorithm in uav navigation using a soc system. In: X SOUTHERN CONFERENCE ON PROGRAMMABLE LOGIC (SPL), 2019. **Proceedings...** [S.l.], 2019. p. 91–96. 33

SCHOTT, C.; PADMANABHA, M.; ROBLER, M.; FROB, D.; HEINKEL, U. Position estimation with bayesian filters by using 3-dimensional environment models. **14th Workshop on Positioning, Navigation and Communications**, 2017. 10, 150

SENANAYAKA, J. S. L.; KHANG, H. V.; ROBBERSMYR, K. G. Multiple classifiers and data fusion for robust diagnosis of gearbox mixed faults. **IEEE Transactions on Industrial Informatics**, 2019. 24

SETIAWAN, J. D.; BUDIMAN, B. A.; ARIYANTO, M.; ANDROMEDA, T.; CHRISMIANTO, D.; AZIZ, M. A. Experimental study on the aerodynamic performance of autonomous boat with wind propulsion and solar power. **6th International Conference on Electric Vehicular Technology**, 2019. 2

SILVA, C. A. O.; GOLTZ, G. A. M.; SHIGUEMORI, E. H.; CASTRO, C. L.; CAMPOS VELHO, H. F.; BRAGA, A. P. Estimation of uav position with use of thermal infrared images. **9th International Conference on Sensing Technology**, 2015. 42

SILVA FILHO, P. F. F. **Identificação automática de referências aplicada à navegação autônoma de vant.** Dissertação (Mestrado em Engenharia Eletrônica e Computação) — Instituto Tecnológico de Aeronáutica, São José dos Campos - SP, 2016. 3, 7, 10, 18, 20, 30, 31, 104

SILVA, W. **Navegação autônoma de vant em período noturno com imagens infravermelho termal**. Dissertação (Mestrado em Computação Aplicada) — Instituto nacional de Pesquisas Espaciais, São José dos Campos - SP, 2016. 7, 19, 24, 29, 39, 42, 43, 56, 104, 108, 111, 113

SILVA, W.; SHIGUEMORI, E. H.; VIJAYKUMAR, N. L.; CAMPOS VELHO, H. F. Estimation of uav position with use of thermal infrared images. In: INTERNATIONAL CONFERENCE ON SENSING TECHNOLOGY, 9., 2015. **Proceedings...** [S.l.], 2015. 33, 111, 113

SINGIRESU, R. **Engineering optimization theory and practice**. [S.l.]: John Wiley, 2009. 55

SRINIVASAN, V.; BHATIA, P.; ONG, S. H. Edge detection using a neural network. In: PATTERN RECOGNITION, 1994. **Proceedings...** [S.l.], 1994. v. 27, n. 12, p. 1653 – 1662. 42

STAMATESCU, G.; STAMATESCU, I.; POPESCU, D.; MATEESCU, C. Sensor fusion method for altitude estimation in mini-uav applications. In: 7TH INTERNATIONAL CONFERENCE ON ELECTRONICS, COMPUTERS AND ARTIFICIAL INTELLIGENCE (ECAI), 2015. **Proceedings...** [S.l.], 2015. 6

STEARNS, P. N. **The industrial revolution in world history**. [S.l.]: Routledge, 2020. 1

SUN, X.; ZHANG, Z.; JI, Y.; YAN, S.; FU, W.; CHEN, Q. Algorithm of ionospheric scintillation monitoring. In: 7TH INTERNATIONAL CONFERENCE ON DIGITAL HOME (ICDH), 2018. **Proceedings...** [S.l.], 2018. 5, 6

TAKAHASHI, H.; TAYLOR, M. J.; SOBRAL, J. H. A.; MEDEIROS, A. F.; GOBBI, D.; SANTANA, D. C. Fine structure of the ionospheric plasma bubbles observed by the oi 6300 and 5577 airglow images. **Advances in Space Research**, v. 27, n. 6, p. 1189–1194, 2001. 5

TAN, Z.; WU, Y.; ZHANG, J. Fused attitude estimation algorithm based on explicit complementary filter and kalman filter for an indoor quadrotor uav. In: CHINESE CONTROL AND DECISION CONFERENCE (CCDC), 2018. **Proceedings...** [S.l.], 2018. 21, 47, 48

URZUA, I.; MUNGUÍA, R.; GRAU, A. Vision-based slam system for mavs in gps-denied environments. **International Journal of Micro Air Vehicles**, v. 9, 2017. 18

VALAVANIS, K. P.; VACHTSEVANOS, G. J. **Handbook of unmanned aerial vehicles**. [S.l.]: Springer Publishing Company, Incorporated, 2014. 3, 69, 84, 97, 98, 122

VALDIVIESO, C.; NOVILLO, F.; GOMEZ, J.; DIK, D. Performance evaluation of channel capacity in wireless sensor networks using ism band in dense urban scenarios. **IEEE Ecuador Technical Chapters Meeting (ETCM)**, 2016. 4

VALENTFX. **Logi pi**. 2020. Disponível em: <[www.valentfx.com/logi-pi/](http://www.valentfx.com/logi-pi/)>. Acesso em: 30 dez. 2020. 71, 125, 126

VIEIRA, L. S. **Planejamento automático de rotas baseado na identificação de marcos aplicado à navegação aérea autônoma**. Dissertação (Mestrado em Computação Aplicada) — Instituto Nacional de Pesquisas Espaciais, São José dos Campos - SP, 2019. 19

VITOR, A.; BRAYAN, R.; EDUARDO, S. R.; MATEUS, T. B.; ELCIO, H. S.; HAROLDO, F.; LUIZ, C.; ANTONIO, P. Combined weightless neural network fpga architecture for deforestation surveillance and visual navigation of uavs. **Engineering Applications of Artificial Intelligence**, 2020. 24, 123, 126

VITOR TORRES, A. M. F.; JAIMES BRAYAN, R. A.; S., E.; T., M.; H., E.; C., H. F.; B., L. C.; P., A. Combined weightless neural network fpga architecture for deforestation surveillance and visual navigation of uavs. **Engineering Applications of Artificial Intelligence**, v. 87, p. 103227, 2020. 20, 33, 55, 58

WANG, L.; HSIEH, W. Toward an improvement of uav-aerial image using non-linear image enhancement. **32nd International Conference on Advanced Information Networking and Applications Workshops**, 2018. 19, 37, 108

WANG, X.; WANG, W. Nonlinear signal-correction observer and application to uav navigation. **IEEE Transactions on Industrial Electronics**, 2019. 8, 10, 21, 50

WEI, W.; TAN, L.; JIN, G.; LU, L.; SUN, C. Computational intelligence for uav autonomous navigation with multisensor data fusion. In: 4TH INFORMATION TECHNOLOGY AND MECHATRONICS ENGINEERING CONFERENCE (ITOEC), 2018. **Proceedings...** [S.l.], 2018. p. 1849–1853. 10, 31

WITTEN, I. H.; FRANK, E.; HALL, A. M. **Data mining: practical machine learning tools and techniques**. [S.l.: s.n.], 2011. 60, 75



XIANG, W.; XINXIN, W.; JIANHUA, Z.; JIANCHAO, F.; XIU, S.; DEJUN, Z. Monitoring the thermal discharge of hongyanhe nuclear power plant with aerial remote sensing technology using a uav platform. In: IEEE INTERNATIONAL GEOSCIENCE AND REMOTE SENSING SYMPOSIUM (IGARSS), 2017.

**Proceedings...** [S.l.], 2017. 17

XILINX. **What is an fpga.** 2020. Disponível em:

<[www.xilinx.com/products/silicon-devices/fpga/what-is-an-fpga.html](http://www.xilinx.com/products/silicon-devices/fpga/what-is-an-fpga.html)>.

Acesso em: 30 dez. 2020. 53

YADAV, S.; SHUKLA, S. Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification. In: 2016 IEEE 6TH INTERNATIONAL CONFERENCE ON ADVANCED COMPUTING, 2016.

**Proceedings...** [S.l.], 2016. 59, 60, 61, 75, 76

YAMAHA MOTORSPORTS. **Precision agriculture rmax.** 2020. Disponível

em: <[www.yamahamotorsports.com/motorsports/pages/precision-agriculture-rmax](http://www.yamahamotorsports.com/motorsports/pages/precision-agriculture-rmax)>.

Acesso em: 30 dez. 2020. 63

YAMAZAKI, Y.; TAMAKI, M.; PREMACHANDRA, C.; PERERA, C. J.; SUMATHIPALA, S.; SUDANTHA, B. H. Victim detection using uav with on-board voice recognition system. **Third IEEE International Conference on Robotic Computing**, 2019. 2, 17, 124

YANG, X.; LIN, D.; ZHANG, F.; SONG, T.; JIANG, T. High accuracy active stand-off target geolocation using uav platform. **IEEE International Conference on Signal, Information and Data Processing**, 2019. 2, 3

YINGKUN, Z. Flight path planning of agriculture uav based on improved artificial potential field method. In: CHINESE CONTROL AND DECISION

CONFERENCE, 2018. **Proceedings...** [S.l.], 2018. 16, 22, 72

YOU, W.; LI, F.; LIAO, L.; HUANG, M. Data fusion of uwb and imu based on unscented kalman filter for indoor localization of quadrotor uav. **20th**

**International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)**, v. 8, 2020. 8, 10

YOUSUF, S.; KADRI, M. B. Robot localization in indoor and outdoor environments by multi-sensor fusion. In: 14TH INTERNATIONAL CONFERENCE ON EMERGING TECHNOLOGIES (ICET), 2019.

**Proceedings...** [S.l.], 2019. 23, 45



ZHAI, X.; ALI, A. A. S.; AMIRA, A.; BENZAALI, F. Mlp neural network based gas classification system on zynq soc. **IEEE Access**, v. 4, p. 8138–8146, 2016. 52

ZHANG, B.; XU, C. Research on uav attitude data fusion algorithm based on quaternion gradient descent. In: INTERNATIONAL CONFERENCE ON COMMUNICATIONS, INFORMATION SYSTEM AND COMPUTER ENGINEERING (CISCE), 2019. **Proceedings...** [S.l.], 2019. 6, 7, 8, 9, 21, 47

ZHILENKOV, A. A.; EPIFANTSEV, I. R. System of autonomous navigation of the drone in difficult conditions of the forest trails. **Conference of Russian Young Researchers in Electrical and Electronic Engineering**, 2018. 2



## **PUBLICAÇÕES TÉCNICO-CIENTÍFICAS EDITADAS PELO INPE**

### **Teses e Dissertações (TDI)**

Teses e Dissertações apresentadas nos Cursos de Pós-Graduação do INPE.

### **Manuais Técnicos (MAN)**

São publicações de caráter técnico que incluem normas, procedimentos, instruções e orientações.

### **Notas Técnico-Científicas (NTC)**

Incluem resultados preliminares de pesquisa, descrição de equipamentos, descrição e ou documentação de programas de computador, descrição de sistemas e experimentos, apresentação de testes, dados, atlas, e documentação de projetos de engenharia.

### **Relatórios de Pesquisa (RPQ)**

Reportam resultados ou progressos de pesquisas tanto de natureza técnica quanto científica, cujo nível seja compatível com o de uma publicação em periódico nacional ou internacional.

### **Propostas e Relatórios de Projetos (PRP)**

São propostas de projetos técnico-científicos e relatórios de acompanhamento de projetos, atividades e convênios.

### **Publicações Didáticas (PUD)**

Incluem apostilas, notas de aula e manuais didáticos.

### **Publicações Seriadas**

São os seriados técnico-científicos: boletins, periódicos, anuários e anais de eventos (simpósios e congressos). Contam destas publicações o Internacional Standard Serial Number (ISSN), que é um código único e definitivo para identificação de títulos de seriados.

### **Programas de Computador (PDC)**

São a seqüência de instruções ou códigos, expressos em uma linguagem de programação compilada ou interpretada, a ser executada por um computador para alcançar um determinado objetivo. Aceitam-se tanto programas fonte quanto os executáveis.

### **Pré-publicações (PRE)**

Todos os artigos publicados em periódicos, anais e como capítulos de livros.