

# Diagnóstico em Tempo Real de Falhas em Drones através das Análises de Sinais de Vibração Baseadas no Caos Usando Densidade e Amplitude de Máximos

Alexandre Magno da Silva Nunes

[msn.alexandre@gmail.com](mailto:msn.alexandre@gmail.com)



CENTRO DE INFORMÁTICA  
UNIVERSIDADE FEDERAL DA PARAÍBA

João Pessoa, 2022

Universidade Federal da Paraíba  
Centro de Informática  
Programa de Pós-Graduação em Informática

Diagnóstico em Tempo Real de Falhas em Drones através das  
Análises de Sinais de Vibração Baseadas no Caos Usando  
Densidade e Amplitude de Máximos

Alexandre Magno da Silva Nunes

Dissertação submetida à Coordenação do Curso de Pós-Graduação em  
Informática da Universidade Federal da Paraíba como parte dos requisi-  
tos necessários para obtenção do grau de Mestre em Informática.

Área de Concentração: Ciência da Computação  
Linha de Pesquisa: Sinais, Sistemas Digitais e Gráficos

Alisson Vasconcelos de Brito  
(Orientador)

João Pessoa, Paraíba, Brasil

©Alexandre Magno da Silva Nunes, 02 de Setembro de 2022

**Catálogo na publicação**  
**Seção de Catalogação e Classificação**

N972d Nunes, Alexandre Magno da Silva.

Diagnóstico em tempo real de falhas em drones através das análises de sinais de vibração baseadas no caos usando densidade e amplitude de máximos / Alexandre Magno da Silva Nunes. - João Pessoa, 2022. 56 f.

Orientação: Alisson Vasconcelos de Brito.  
Dissertação (Mestrado) - UFPB/CI.

1. Informática. 2. Drones. 3. Diagnóstico de falhas. 4. Análise de sinais. 5. Aprendizagem de máquina. I. Brito, Alisson Vasconcelos de. II. Título.

UFPB/BC

CDU 004(043)



**MINISTÉRIO DA EDUCAÇÃO  
UNIVERSIDADE FEDERAL DA PARAÍBA**

**ATA Nº 8 / 2022 - CI-DSC (18.58)**

**Nº do Protocolo: 23074.087513/2022-36**

**João Pessoa-PB, 14 de Setembro de 2022**

**Ata da Sessão Pública de Defesa de Dissertação de Mestrado de Alexandre Magno da Silva Nunes, candidato ao título de Mestre em Informática na Área de Sistemas de Computação, realizada em 31 de agosto de 2022.**

Aos trinta e um dias do mês de agosto, do ano de dois mil e vinte e dois, às quatorze horas, no Centro de Informática da Universidade Federal da Paraíba, em Mangabeira, reuniram-se os membros da Banca Examinadora constituída para julgar o Trabalho Final do Sr. Alexandre Magno Da Silva Nunes, vinculado a esta Universidade sob a matrícula nº 20201003579, candidato ao grau de Mestre em Informática, na área de “Sistemas de Computação”, na linha de pesquisa “Sinais, Sistemas Digitais e Gráficos”, do Programa de Pós-Graduação em Informática, da Universidade Federal da Paraíba. A comissão examinadora foi composta pelos professores: Alisson Vasconcelos de Brito (PPGI-UFPB), Orientador e Presidente da Banca, Tiago Pereira do Nascimento (PPGI-UFPB), Examinador Interno, Abel Cavalcante Lima Filho (UFPB), Examinador Externo ao Programa, Jorge Gabriel Gomes de Souza Ramos (UFPB), Examinador Externo ao Programa. Dando início aos trabalhos, o Presidente da Banca cumprimentou os presentes, comunicou a finalidade da reunião e passou a palavra ao candidato para que ele fizesse a exposição oral do trabalho de dissertação intitulado “Diagnóstico em Tempo Real de Falhas em Drones através das Análises de Sinais de Vibração Baseadas no Caos Usando Densidade e Amplitude de Máximos”. Concluída a exposição, o candidato foi arguido pela Banca Examinadora que emitiu o seguinte parecer: “aprovado”. Do ocorrido, eu, Fernando Menezes Matos, Coordenador do Programa de Pós-Graduação em Informática, lavrei a presente ata que vai assinada por mim e pelos membros da banca examinadora.

João Pessoa, 31 de agosto de 2022.

*(Assinado digitalmente em 14/09/2022 07:38 )*

**ABEL CAVALCANTE LIMA FILHO  
PROFESSOR DO MAGISTERIO SUPERIOR  
Matrícula: 1753873**

*(Assinado digitalmente em 14/09/2022 06:46 )*

**ALISSON VASCONCELOS DE BRITO  
PROFESSOR DO MAGISTERIO SUPERIOR  
Matrícula: 2579537**

*(Assinado digitalmente em 14/09/2022 08:14 )*

**FERNANDO MENEZES MATOS  
COORDENADOR DE CURSO  
Matrícula: 1887274**

*(Assinado digitalmente em 14/09/2022 10:15 )*

**JORGE GABRIEL GOMES DE SOUZA RAMOS  
PROFESSOR DO MAGISTERIO SUPERIOR  
Matrícula: 1572287**

*(Assinado digitalmente em 14/09/2022 08:24 )*

**TIAGO PEREIRA DO NASCIMENTO  
PROFESSOR DO MAGISTERIO SUPERIOR  
Matrícula: 2042972**

Para verificar a autenticidade deste documento entre em <https://sipac.ufpb.br/documentos/> informando seu número: **8**, ano: **2022**, documento(especie): **ATA**, data de emissão: **14/09/2022** e o código de verificação: **c90b565190**

## **Agradecimentos**

À Deus, com a sua infinita bondade, por me permitir chegar até aqui.

À minha Mãe, meu Pai, minhas irmãs, minha namorada Carol, meus amigos e sobretudo às minhas avós Severina e Gessi, por todo o amor, apoio emocional, educação e tudo que é digno a qualquer pessoa.

Aos amigos do LaSER Theo, Matheus e Nico que tiveram uma contribuição direta no projeto.

Ao amigo, professor, orientador e mentor Alisson, por acreditar no potencial do projeto e por ser o exemplo de profissional que me esforçarei para ser um alcançar.

## Resumo

O Veículo Aéreo Não Tripulado (VANT) se popularizou ao longo dos anos em diversas atividades, sejam comerciais ou recreativas. Porém, os drones ainda não possuem total confiança no que se refere a segurança. Diante disso, surge a necessidade de criação de técnicas de identificação de falhas que se adequem às limitações encontradas. Este trabalho tem como objetivo a detecção e diagnóstico de falhas em drones em tempo real e propõe uma arquitetura para identificação de falhas mecânicas. A abordagem proposta no trabalho utiliza as técnicas de análises de sinais baseadas no caos usando a densidade de máximos (SAC-DM) e amplitude de máximos (SAC-AM). Para atingir os resultados foram capturados dados de um drone quadróptero e aplicadas as técnicas e algoritmos classificadores. No SAC-DM foram alcançados desempenhos de 91% com o KNN e 90% com o SVM, já no SAC-AM o resultado foi de 100% com o KNN e 98% com o SVM. Também foi possível validar o sistema diagnosticando falhas em tempo real.

**Palavras-chave:** VANT, Drones, Diagnóstico de Falhas, Análise De Sinais, Aprendizagem de Máquina

## **Abstract**

The Unmanned Aerial Vehicle (UAV) has become popular over the years in various activities, whether commercial or recreational. However, drones still do not have complete confidence when it comes to safety. Therefore, there is a need to create fault identification techniques that fit the limitations found. This work aims to detect and diagnose drone failures in real time and proposes an architecture to identify mechanical failures. The approach proposed in the work uses chaos-based signal analysis techniques using maximum density (SAC-DM) and maximum amplitude (SAC-AM). To achieve the results, data were captured from a quadcopter drone and applied classification techniques and algorithms. With SAC-DM, performances of 91% were achieved and with SAC-AM 100%. It was also possible to validate the system by diagnosing faults in real time.

**Keywords:** UAV, Drones, Fault Diagnosis, Signal Analysis, Machine Learning

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Objetivos . . . . .	2
1.1.1	Objetivo geral . . . . .	2
1.1.2	Objetivos específicos . . . . .	2
1.2	Estrutura da Dissertação . . . . .	3
<b>2</b>	<b>Trabalhos Relacionados</b>	<b>4</b>
<b>3</b>	<b>Fundamentação Teórica</b>	<b>8</b>
3.1	Falhas mais comuns em VANTs e sistemas de detecção e acomodação de falhas	8
3.2	Sistemas Caóticos . . . . .	9
3.2.1	Análise de Sinal Baseada no Caos usando Densidade de Máximos (SAC-DM) . . . . .	10
3.2.2	Análise de Sinal Baseada no Caos usando Amplitude de Máximos (SAC-AM) . . . . .	11
3.3	Aprendizado de Máquina . . . . .	11
3.4	Serviços web . . . . .	13
<b>4</b>	<b>Materiais e Métodos</b>	<b>15</b>
4.1	Arquitetura do sistema proposto . . . . .	15
4.2	Aquisição do conjunto de dados . . . . .	17
4.3	Técnicas utilizadas . . . . .	20
4.3.1	Implementação da SAC-DM . . . . .	20
4.3.2	Implementação da SAC-AM . . . . .	21
4.3.3	K-Vizinhos mais Próximos . . . . .	21

---

4.3.4	Máquinas de Vetor de Suporte . . . . .	22
4.4	Ferramentas utilizadas . . . . .	22
4.5	Avaliação de desempenho . . . . .	23
4.5.1	Matriz de confusão . . . . .	23
4.5.2	Acurácia . . . . .	24
<b>5</b>	<b>Resultados</b>	<b>25</b>
5.1	Configuração dos teste . . . . .	25
5.2	Programação Sistema Embarcado . . . . .	26
5.3	Implementação do Servidor e do Serviço de Análise . . . . .	26
5.4	Implementação do Aplicativo Web . . . . .	26
5.5	Teste e Validação do sistema ( <i>off-line</i> ) . . . . .	28
5.6	Teste e Validação do sistema ( <i>on-line</i> ) . . . . .	35
<b>6</b>	<b>Conclusão</b>	<b>37</b>

# Lista de Símbolos

**ANAC** : *Agência Nacional de Aviação Civil*

**CCC** : *Coeficiente de Comprimento de Correlação*

**HTTP** : *Hypertext Transfer Protocol*

**IMU** : *Inertial Measurement Unit*

**IoT** : *Internet Of Things*

**KNN** : *K-Nearest Neighbors*

**LASER** : *Laboratório de Engenharia de Sistemas e Robótica*

**ML** : *Machine Learning*

**MQTT** : *Message Queuing Telemetry Transport*

**REST** : *Representational State Transfer*

**RPA** : *Remotely-Piloted Aircraft*

**SAC-AM** : *Analysis Based on Chaos using Amplitude of Maxima*

**SAC-DM** : *Analysis Based on Chaos using Density of Maxima*

**SVM** : *Support-Vector Machine*

**UFPB** : *Universidade Federal da Paraíba*

**VANT** : *Veículo Aéreo Não Tripulado*

# Lista de Figuras

4.1	Arquitetura do sistema proposto. . . . .	16
4.2	Diagrama de blocos do fluxo dos dados. . . . .	17
4.3	Quadricoptero utilizado na aquisição de dados. . . . .	18
4.4	Espaço utilizada para realização dos experimentos. . . . .	19
4.5	Hélice desbalanceada e hélice balanceada. . . . .	19
5.1	Tela inicial da aplicação. . . . .	27
5.2	Tela para realizar coletas. . . . .	27
5.3	Tela para realizar diagnósticos. . . . .	28
5.4	Comparativo entre os sinais brutos do voo normal, voo com falha 2 e voo com falha 3, para os três eixos do acelerômetro. . . . .	29
5.5	Comparativo entre os sinais com SACDM do voo normal, voo com falha 2 e voo com falha 3, para os três eixos do acelerômetro. . . . .	29
5.6	Comparativo entre os sinais com SACAM do voo normal, voo com falha 2 e voo com falha 3, para os três eixos do acelerômetro. . . . .	30
5.7	Desempenho do algoritmo KNN para os sinais brutos. . . . .	31
5.8	Desempenho do algoritmo KNN para o SAC-DM aplicado. . . . .	31
5.9	Desempenho do algoritmo KNN para o SAC-AM aplicado. . . . .	32
5.10	Desempenho do algoritmo SVM para os sinais brutos. . . . .	33
5.11	Desempenho do algoritmo SVM para o SAC-DM aplicado. . . . .	33
5.12	Desempenho do algoritmo SVM para o SAC-AM aplicado. . . . .	34
5.13	Tela de diagnósticos apresentando alguns resultados . . . . .	35

# Lista de Tabelas

3.1	Diferenças entre SOAP e REST. . . . .	14
4.1	Cenário dos Voos . . . . .	19
4.2	Matriz de confusão . . . . .	24
5.1	Desempenhos dos algoritmos para cada técnica. . . . .	34
5.2	Desempenhos final do experimento on-line. . . . .	36

# Lista de Algoritmos

1	SAC-DM . . . . .	20
2	SAC-AM . . . . .	21

# Capítulo 1

## Introdução

Os equipamentos que utilizam o espaço aéreo sem a necessidade de um piloto a bordo podem receber várias denominações, uma delas é a de Veículo Aéreo Não Tripulado (VANT), que é a tradução de Unmanned Aerial Vehicle (UAV), e é bastante encontrada na literatura. Porém esse termo se tornou ultrapassado para a Organização da Aviação Civil Internacional (ICAO, 2020) que adotou a expressão Sistemas de Aeronave Pilotada Remotamente, traduzida de Remotely Piloted Aircraft Systems (*RPAS/RPA*) (DECEA, 2020). “Drones” também é uma denominação bastante popular, mas se trata de um termo genérico, sem reconhecimento na ciência ou na legislação.

No Brasil, pelo regulamento da Agência Nacional de Aviação Civil (ANAC), o termo mais adequado também é *RPA*, e se refere àquelas aeronaves pilotadas remotamente utilizadas para fins experimentais, comerciais ou institucionais. O aeromodelo também é uma denominação prevista no regulamento da ANAC, diferenciando-se da *RPA* pela sua finalidade de recreação (ANAC, 2020).

Pelos registros históricos os primeiros VANT surgiram em 1849, se tratavam de balões austríacos que foram utilizados para bombardear a cidade de Veneza, na Itália (COOK, 2007). Em 1863, foi registrada uma patente nos Estados Unidos de um balão bombardeiro sem tripulante (SANTOS, 2017). A primeira aeronave não tripulada surgiu em 1916, e com o passar dos anos o desenvolvimento desses equipamentos passou por constantes evoluções realizadas por forças militares (PRISACARIU et al., 2017).

O Sistema de Aeronave Remotamente Pilotada se popularizou devido à suas inúmeras possibilidades de aplicações e seu baixo custo operacional. Entretanto, existem algumas res-

trições impostas por agências de aviação para utilização desse tipo de veículo (MORALES; PAEZ; ARANGO, 2015). Isso se dá pelo fato de serem considerados sistemas críticos de segurança, isto é, por causa de uma falha ou mau funcionamento, podem causar ferimentos graves ou até mesmo a morte de pessoas (KING; BERTAPELLE; MOSES, 2005). Procedimentos de certificação de confiabilidade são extremamente necessários para sistemas como esses. A aviação tripulada, que também é um desses sistemas, possui rigorosos requisitos de confiabilidade que permitem que uma aeronave funcione em média 109 horas sem apresentar nenhuma falha catastrófica (DALAMAGKIDIS; VALAVANIS; PIEGL, 2009). Porém nem todas as abordagens utilizadas na aviação tripulada podem ser aplicadas na aviação não tripulada, em razão das limitações de recursos dos RPA's (FREEMAN, 2013).

Diante disso, surge a necessidade de criação de técnicas de identificação de falhas que se adequem às limitações encontradas. As metodologias encontradas para reconhecimento dos defeitos nos VANT's normalmente são divididas em três abordagens: conhecimento, processamento de sinal ou baseadas em modelo. Este trabalho tem como objetivo a detecção e diagnóstico de falhas em drones em tempo real e propõe uma estratégia para identificação de falhas mecânicas em hélices. A abordagem proposta no trabalho utiliza as técnicas de análises de sinais baseadas no caos usando a densidade de máximos (SAC-DM) e amplitude de máximos (SAC-AM).

## **1.1 Objetivos**

### **1.1.1 Objetivo geral**

Desenvolver uma arquitetura de um sistema de diagnóstico de falhas em drones, que possa ser utilizado em tempo real, e que auxilie na definição de estratégias de segurança do voo.

### **1.1.2 Objetivos específicos**

- Aplicar no método de diagnóstico a análise da densidade de máximos;
- Aplicar no método de diagnóstico a análise da amplitude de máximos;
- Demonstrar a viabilidade técnica e financeira do sistema aplicado a um drone quadri-

coptero em diferentes condições controladas.

## **1.2 Estrutura da Dissertação**

Este trabalho está organizado da seguinte maneira: o capítulo 1 apresenta a contextualização do problema a ser explorado, bem como os objetivos geral e específicos. No capítulo 2 são apresentados alguns trabalhos relacionados, para demonstrar o grau de conhecimento do assunto e os pontos de aderência ao presente trabalho. No capítulo 3 é apresentado a fundamentação teórica com os principais conceitos necessários para compreensão do projeto. No capítulo 4 são descritos os materiais e métodos utilizados para a obtenção dos resultados. No capítulo 5 são apresentados os resultados e descritos os processos para alcançá-los. Em seguida, no capítulo 6 é realizada a conclusão do trabalho destacando as principais contribuições, análise dos resultados obtidos e sugestões para trabalhos futuros.

# Capítulo 2

## Trabalhos Relacionados

Este capítulo tem como objetivo fazer uma iniciação ao estado da arte, para demonstrar o grau de conhecimento atual sobre o objeto de estudo deste trabalho, destacando os pontos de aderência com o tema desta pesquisa. Os trabalhos que serão apresentados a seguir mostram algumas abordagens para realização de detecção e diagnóstico de falhas ( FDD - Fault Detection and Diagnostics ) em diferentes tipos de drones e dispositivos rotativos.

No trabalho de Bondyra et al. (2017), foi apresentado um método para detecção de falhas físicas em hélices de drones multirotores. O método consistiu em um algoritmo baseado em processamento de sinais e aprendizagem de máquina. O sinal utilizado foi o de aceleração, proveniente de um sensor IMU (Inertial Measurement Unit), que foram armazenados e aplicados em seguida no classificador Support Vector Machine (SVM). Para extração de características do sinal, Bondyra et al. (2017) utilizou: transformada rápida de Fourier (FFT), decomposição discreta do sinal em pacotes wavelet e o cálculo da potência média do sinal em bandas de frequência. A taxa de classificações corretas ficou entre 80% a 90%, com a FFT obtendo o melhor resultado.

Baskaya, Bronz e Delahaye (2017), na mesma linha de Bondyra et al. (2017), propuseram uma metodologia para classificar as condições de voo utilizando o método de classificação SVM, para classificar as condições de voo com defeito e normais. Os dados utilizados também foram medições provenientes de um giroscópio e acelerômetro. A falha considerada no trabalho foi a perda de eficácia nas superfícies de controle do drone. Os resultados deste trabalho também mostraram que o SVM oferece um bom desempenho de classificação das falhas analisadas pelos autores.

---

Em Pedotti et al. (2019), foi apresentado um dispositivo de baixo custo, que tem como objetivo mensurar sinais de vibrações de um acelerômetro aplicado em máquinas rotativas que operam em velocidades de até 4000 RPM. No trabalho foi utilizado um microcontrolador ESP 32, que além de receber os dados do sensor, também foi usado para calcular a FFT do sinal a cada 1,25s e transmissão dos dados para um servidor em nuvem. Para possibilitar o diagnóstico das perturbações inseridas, foram utilizadas técnicas como regressão logística, linear SVM e redes neurais artificiais.

Na abordagem de López-Estrada et al. (2021) foi realizada a captura de dados de voo de um drone quadricoptero. Os voos foram divididos em duas etapas. A primeira o drone voou em condições normais. Na segunda, o drone voou com uma das hélices cortada para que causasse uma redução no empuxo gerado pelos rotores. Os dados utilizados foram oriundos do giroscópio e acelerômetro do próprio drone. Para a extração de característica foi empregada a Análise de Componentes Principais (PCA). Em seguida foi aplicados os algoritmos de aprendizagem de máquina k-vizinho mais próximo (k-NN) e de máquina de vetores de suporte (SVM), os quais obtiveram desempenho de 89,6% (k-NN) e 92,4% (SVM), respectivamente.

Em Zhang et al. (2021) também é proposto um método para detecção e identificação de falhas utilizando sinais de vibração, similarmente à Bondyra et al. (2017). O método de Zhang et al. (2021), coleta os dados de aceleração, extrai características do sinal com decomposição do pacote wavelet e integra com o modelo de rede neural de memória de longo e curto prazo (LSTM). O trabalho de Zhang et al. (2021) obteve 96% de precisão, na sua melhor combinação de parâmetros, onde foram utilizados os três eixos (X, Y e Z) do acelerômetro aplicado.

Em Hong e Lee (2021), foi posicionado um acelerômetro ao um dos motores conectado a uma das hélices do drone, para captação do sinal de vibração em voos considerados normais e anormais. Para realizar o diagnóstico em cima dos dados, os autores empregaram os métodos de rede neural recorrente (RNN) e LSTM, onde foi realizada uma comparação entre os métodos. Como resultado da simulação comparativa, tanto o RNN quanto o LSTM tiveram bons resultados, entretanto, o RNN apresentou o melhor resultado.

Em Ghalamchi, Jia e Mueller (2019), também é proposto uma abordagem para detecção e diagnóstico de desgastes em motores e hélices de drones, por meio do sinal de saída de um

---

acelerômetro. A abordagem é baseada no filtro de Kalman, onde estima-se um desequilíbrio para cada hélice e realiza-se a detecção diretamente no domínio no tempo, demandando menos recursos computacionais. Os experimentos na proposta apresentada mostram uma confiável detecção para uma hélice com defeito.

Zhong e Xin (2019) propuseram a combinação do filtro de Kalman sem cheiro e o fator de desvanecimento subótimo, com o intuito de reconhecer falhas em UAVs. O método estima e prediz o estado do sistema por meio de um modelo discretizado. No trabalho foram usados parâmetros gerais, dados aerodinâmicos e dados do motor. O experimento apresentou boa capacidade para reconhecimento de falha em atuadores e sensores, além de apresentar um desempenho rápido e eficaz.

O trabalho de Bektash e La Cour-Harbo (2020) testou a hipótese de que falhas em drones podem ser detectadas utilizando o método de Welck, também conhecido como método do periodograma. No trabalho, as estimativas de espectro de potência de ciclos segmentados são indicadores de anomalias. Os resultados indicaram que um processo sistemático de coleta pode ser usado para avaliação da integridade mecânica do drone.

Medeiros et al. (2018) propõe um método de detecção e diagnóstico de falhas em motores de corrente contínua sem escovas (BLDC), baseado na análise do comportamento caótico. No trabalho, o objeto de estudo é o sinal de corrente elétrica desses motores no domínio do tempo. A técnica proposta por Medeiros et al. (2018) é chamada de Análise de Sinais Baseada no Caos usando Densidade de Máximo (SAC-DM), e consiste na análise das médias da densidade dos máximos das amostras. Os experimentos do trabalho foram realizados numa bancada e os resultados mostraram que é possível detectar falhas no motor.

Em Veras et al. (2019), a técnica SAC-DM foi aplicada a partir de sinais sonoros, para identificar e diagnosticar falhas em motores BLDC. No trabalho, é utilizada a Análise wavelet multiresolução a fim de separar a componente caótica do sinal sonoro. Para validação do método proposto, foram realizados testes em uma bancada montada com um motor BLDC e uma hélice, os quais foram submetidos às condições de falhas. No experimento, foi utilizado para captura dos sinais sonoros um sistema desenvolvido sob a plataforma Arduino e um microfone condensador. A abordagem demonstrou boa capacidade em detectar a velocidade do motor e problemas de excentricidade, alcançando desempenho superior a 90%.

No trabalho de Souza et al. (2020), é apresentado um experimento utilizando um drone de

pequeno porte, com o objetivo de aplicar a técnica de SAC-DM para detecção de falhas. No experimento, é utilizado um circuito elaborado pelos autores, que foi embarcado no próprio drone. Esse circuito foi capaz de coletar os valores de SAC-DM calculados a partir de dados de vibração, capturados por meio de um acelerômetro. Também foi desenvolvido um serviço de banco de dados, para receber os dados enviados pelo circuito embarcado e armazená-los para visualização e análise posterior. Os voos realizados no experimento combinaram três condições: hélices balanceadas, uma ou duas hélices desbalanceadas e o centro de gravidade deslocado. Para realizar a detecção, o algoritmo desenvolvido testou se um valor estava em um intervalo fixo ou não. Os resultados apresentados foram pertinentes para detecção das falhas propostas e apresentou desempenho de 82,75%. Porém os autores apontaram uma limitação para realizar o diagnóstico das falhas.

Diante do cenário de pesquisas pode-se observar uma grande preocupação da comunidade científica com a segurança nos RPAS. Os trabalhos encontrados mostram diferentes abordagens para realizar detecção e diagnóstico das possíveis falhas apresentadas nos drones. Também foi possível observar que não há um consenso sobre qual abordagem é mais adequada, porém todas que foram apresentadas se mostraram relevantes para o tema.

# Capítulo 3

## Fundamentação Teórica

### 3.1 Falhas mais comuns em VANTs e sistemas de detecção e acomodação de falhas

Os tipos de falhas mais comuns em VANTs, podem surgir na carcaça (estrutura física), no atuador ou nos sensores (CORK; WALKER; DUNN, 2005). As falhas na carcaça normalmente são danos estruturais, decorrentes de quedas, colisões, desbalanceamento de hélices ou defeitos de fabricação. A falhas nos atuadores resultam de problemas mecânicos. Estes tipos de falhas são inviáveis para utilizar técnicas de redundância de acomodação e precisam de adaptações no sistema de controle. Outro tipo de falha recorrente é nos sensores, que podem oferecer riscos irrecuperáveis ao voo. Existe uma grande importância de combinar um conjunto de técnicas para identificar os três tipos de falhas, porém o foco deste trabalho é na detecção e acomodação na estrutura.

As Metodologias para detectar e classificar as falhas normalmente são divididas em três abordagens: conhecimento, processamento de sinal ou baseadas em modelo (CORK; WALKER; DUNN, 2005). A abordagem de conhecimento pode utilizar ferramentas como redes neurais ou lógica de decisão difusa, entre outras técnicas de inteligência artificial. A abordagem de processamento de sinais, busca características como espectro e informações estatística do sinal para analisar a existência de falha. Já à abordagem baseada em modelo, fazem estimativas das medidas dos valores a partir de condições de sinais de erro. Para avaliar o desempenho de uma técnica de detecção de falhas, é calculado o percentual de medições bem

sucedidas, alarmes falsos e alarmes perdidos. Quando o resultado do diagnóstico é correto, o caso é classificado como bem sucedido. Quando uma falha é alertada sem a existência do problema, denomina-se alarme falso. Por último, quando uma falha ocorre mas não é detectada, é considerado um alarme perdido. Os sistemas de detecção de falhas para veículos aéreos não tripulados, tem como objetivo final permitir que o drone permaneça em voo com desempenho aceitável, até que sua missão possa ser concluída ou que se recupere da falha. Porém, por mais confiável que um sistema possa ser, ele deve sempre ser apoiado por manutenção adequada para maior segurança.

## 3.2 Sistemas Caóticos

A teoria do Caos, também conhecida como ciência da complexidade, é um dos temas que mais se expandiu nas últimas décadas (WILLIAMS, 1997). Caos é conhecido no senso comum como desordem, confusão e desequilíbrio. Para o sentido científico, o caos implica que qualquer alteração infinitesimal dentro de um sistema organizado pode ocasionar consequências imprevisíveis. Cambel (1993) defende que a natureza complexa das composições, formas e funções de tudo que cerca o universo, torna impossível a classificação de padrões. Isso se deve pelo fato da relação entre a causa e o efeito não serem proporcionais. Do mesmo modo que um grande efeito pode ter pequenas consequências, um pequeno efeito pode ocasionar um grande resultado. Esses eventos são chamados de não lineares e normalmente constituem a complexidade.

Em estudos realizados na década de 90 foi percebido que as ferramentas da teoria do caos podem ser utilizadas na compreensão, manipulação e controle de vários tipos de sistemas (DITTO, 1995). Porém, não existe uma definição padrão que caracterize a presença do caos em um determinado sistema. Apesar disso, existem características típicas que podem revelar o comportamento caótico. Essas características incluem:

- **Não-linearidade:** Não há como um sistema ser caótico sem atender esta característica;
- **Determinismo:** Os sistemas se desenvolvem de um momento para o outro, seguindo regras determinísticas, onde o sistema presente depende do passado;

- **Sensibilidade às condições iniciais:** Pequenas mudanças no estado inicial podem levar a um comportamento radicalmente diferente em seu estado final;
- **Imprevisibilidade de longo prazo:** É praticamente impossível prever o comportamento do sistema ao longo prazo devido à sensibilidade às condições iniciais.

O caos determinístico significa que o comportamento futuro de um sistema é totalmente determinado pelas condições iniciais, sem que existam componentes aleatórias. No entanto, devido à extrema sensibilidade às condições iniciais, as previsões de longo prazo desse comportamento podem se tornar inviáveis.

### 3.2.1 Análise de Sinal Baseada no Caos usando Densidade de Máximos (SAC-DM)

A Análise de Sinais Baseada no Caos usando Densidade de Máximo (SAC-DM), é um procedimento que se fundamenta na medição da desordem das partículas de um sistema físico, que é conhecido como Entropia, na termodinâmica e também se baseia na teoria do caos (MEDEIROS et al., 2018). Neste método o comportamento de equilíbrio cíclico é a questão central. Uma das possíveis aplicações deste método é a identificação do comportamento caótico em sinais de vibração capturados em motores de corrente contínua sem escova, que são muito utilizados em UAVs.

No trabalho de Medeiros et al. (2018) foi demonstrada uma fórmula simplificada, deduzida pelas propriedades caóticas dos sistemas estocásticos, que considera apenas a densidade máxima dos sinais. A fórmula se apresenta da seguinte forma:

$$\tau = \frac{1}{\hat{6}(\rho)} \quad (3.1)$$

Onde  $\tau$  representa o Comprimento de Correlação e  $\rho$  a Densidade de Máximos.

O método é uma nova alternativa para se obter o Coeficiente de Comprimento de Correlação (Correlation Length Coefficient), a partir da computação da densidade de picos de um sinal. O Coeficiente de Comprimento de Correlação é um método que utiliza a correlação probabilística para analisar o comportamento caótico. Este comportamento é quantificado através do coeficiente de correlação. A solução calcula a média da densidade dos máximos

das amostras, o que permite uma estimativa das propriedades de um determinado sistema em uma série temporal única e curta.

Com isso, podemos observar que a implementação do método irá produzir algoritmos de baixa complexidade, o que viabiliza seu uso em sistemas embarcados com poder computacional reduzido.

### 3.2.2 Análise de Sinal Baseada no Caos usando Amplitude de Máximos (SAC-AM)

Neste trabalho, é apresentada outra análise baseada no caos, denominada Análise de Sinal baseada em Caos usando Amplitude de Máxima (SAC-AM). A amplitude dos picos em um sinal periódico é um marco crucial para estudar o caos (LAKSHMANAN; RAJASEKAR, 2003) (SHIMIZU; SEKIKAWA; INABA, 2011). É esperado que a amplitude dos sinais dos sistemas caóticos também mantenha o mesmo comportamento que a densidade. Além disso, espera-se que sejam ainda mais sensíveis a distúrbios externos.

O sinal é expresso como uma função  $f(t)$ . O SAC-AM gera outro sinal  $S(w_i)$ , onde  $w_i$  é a  $i$ -ésima janela de tamanho  $w$ .  $p_k$  é uma lista com todos os picos em  $f(t)$ . Então,  $S(w_i) = S_0, S_1, S_2, \dots, S_n$ , onde:

$$S_i = \sum_k^{w_i} \frac{f(p_k)}{w_i} \quad (3.2)$$

## 3.3 Aprendizado de Máquina

Com a crescente quantidade de dados disponíveis de diferentes origens, foi necessário criar ferramentas e técnicas que pudessem auxiliar na análise e extração de informações de forma inteligente. O aprendizado de máquina, do inglês Machine Learning (ML), tem se tornado um dos pilares da computação para produzir resultados a partir de dados de entrada (EL NAQA; MURPHY, 2015). Um dos indicadores que demonstram o sucesso das técnicas de aprendizado de máquina é a quantidade de áreas que aplicam seus métodos, tais como: visão computacional, engenharia, finanças, ciências biológicas e médicas, entretenimento, etc (SHINDE; SHAH, 2018).

Na literatura pode ser encontrada algumas definições formais para aprendizado de máquina. A definição mais citada é a de Samuel (1959) que diz que é um “campo de estudo que dá aos computadores a habilidade de aprender sem serem explicitamente programados”.

A técnica de ML explora diferentes algoritmos, que podem aprender com seus erros e fazerem previsões para resolver problemas de dados (BONACCORSO, 2017). O tipo da aprendizagem e o algoritmo a ser aplicado dependerá do problema que se deseja resolver, do número de variáveis, do modelo que melhor se adapta, etc. Alguns dos tipos de aprendizagem de máquina são: aprendizagem supervisionada, aprendizagem não supervisionada, aprendizagem semi-supervisionada e aprendizagem por Reforço (MAHESH, 2020).

- **Aprendizagem supervisionada:** Neste tipo de aprendizagem o sistema tenta encontrar associações e padrões entre a entrada e a saída, que já são previamente conhecidas, para realizar previsões. Os algoritmos de aprendizagem supervisionada precisam de assistência externa. Entre alguns dos algoritmos que seguem esta abordagem, estão:
  - Árvore de decisão;
  - Navie Bayes;
  - Máquina de vetor de suporte (Support Vector Machine - SVM );
  - k-vizinhos mais próximos ( k-nearest neighbors algorithm - k-NN).
- **Aprendizagem não supervisionada:** Nesta abordagem a entrada e a saída não são conhecidas, dessa forma, a aprendizagem ocorre de forma implícita em conjuntos de dados não rotulados. Os algoritmos por conta própria buscam identificar padrões para rotular os dados. Entre alguns dos algoritmos que seguem esta abordagem, estão:
  - K-Means Clustering;
  - Propagação de afinidade;
  - Redução de dimensionalidade;
  - Modelos de distribuição.
- **Aprendizagem semi-supervisionada:** Esta abordagem sugere que em grandes conjuntos de dados, existe uma parcela rotulada e outra não rotulada. Dessa forma os algoritmos desta abordagem aplicam uma combinação das duas abordagens anteriores,

para melhorar o desempenho do sistema. Alguns algoritmos que podem ser citados, são:

- SVM transdutivo;
  - COP-k-means;
  - SEEDED-k-means;
- **Aprendizagem por Reforço:** Nesta abordagem o aprendizado ocorre com base na tentativa e erro. Para que se chegue ao objetivo final, o sistema vai sendo recompensado ou penalizado à medida em que executa suas ações. O objetivo é maximizar as recompensas. Alguns algoritmos desta abordagem são:
- Avaliação de Política iterativa;
  - Iteração de Valor;
  - Q-learning;

## 3.4 Serviços web

Os serviços na web utilizam o modelo cliente-servidor para distribuir seus recursos, para os seus requerentes (MAGRI, 2013). O cliente está em uma das pontas da conexão, e é representado pelos programas de acesso à Web disponíveis nos dispositivos conectados à internet. Na outra ponta da conexão estão os servidores, que atendem às solicitações enviadas pelos clientes. A comunicação segue os padrões HTTP, que é um protocolo da camada de aplicação, e enviado sobre o protocolo TCP/IP, que é um protocolo da camada de transmissão (TANENBAUM, 2003). A comunicação no padrão HTTP funciona através de trocas de mensagens, que podem ser requisições (request) ou respostas (responses). O fluxo da comunicação é projetado para ser simples, onde basicamente o cliente envia uma mensagem de requisição ao servidor, e este retorna uma mensagem de resposta.

O corpo de uma requisição é composto por:

- Um método HTTP, que tem entre os mais comuns os verbos: GET, POST, DELETE, PUT;

- O caminho (URL) do recurso a ser disponibilizado; A versão do protocolo HTTP;
- Um cabeçalho contendo informações adicionais para o servidor, que pode ser opcional;
- Um corpo de dados, para métodos como POST e PUT.

Já a estrutura de uma resposta do servidor segue o seguinte padrão:

- A versão do protocolo HTTP;
- Um código de status: (1xx) Informação, (2xx) Sucesso, (3xx) Redirecionamento, (4xx) Erro no cliente, (5xx) Erro no servidor;
- A descrição do status;
- O corpo de dados solicitados, a depender do método na requisição.

Tabela 3.1: Diferenças entre SOAP e REST.

SOAP	REST
Utiliza XMLs para a transferência de objetos entre aplicações;	É mais comum usar JSON;
O objeto, é serializado para XML e desserializado para o formato original;	Utiliza o HTTP puro estabelecendo algumas convenções;
Existem mais regras que o REST, tornando-o mais burocrático.	Os serviços Web em conformidade com estilo arquitetural REST, são denominados RESTful.

Uma aplicação Web pode ser vista como uma coleção de páginas, onde o conteúdo renderizado é dinâmico, montado através de requisições a um servidor que faz consultas em um banco de dados (JAZAYERI, 2007). Normalmente, são esses tipos de aplicação que consomem recursos de serviços web. Entre os padrões mais comuns de serviços web, podemos destacar o SOAP e o REST (TIHOMIROVS; GRABIS, 2016). Na tabela 3.1 são listadas algumas características e diferenças entre eles.

# Capítulo 4

## Materiais e Métodos

### 4.1 Arquitetura do sistema proposto

Para a análise em tempo real, é proposta uma arquitetura que será composta por um sistema embarcado, um servidor e um Serviço de Análise.

O sistema embarcado é composto por um kit de ferramentas de desenvolvimento de código aberto, incluindo hardware e soluções de IoT, desenvolvido pela M5Stack (M5STACK, 2020). O kit a ser utilizado possui uma ESP32, que contém Wi-Fi e modo duplo Bluetooth, IMU de 6 eixos, entre outros recursos. Além disso, existe uma ampla biblioteca aberta, que facilita o processo de desenvolvimento. O kit foi acoplado no drone e assumiu as funções de aquisição e transmissão de dados.

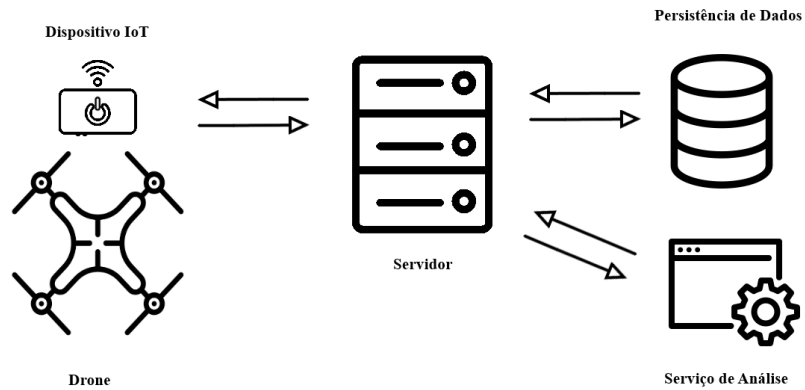
O servidor proposto na arquitetura funciona localmente, a princípio, e atende às requisições do sistema embarcado, para receber os dados enviados. Após recebidos os dados, o servidor aplica uma das técnicas de análises de sinais de vibração baseadas no caos, para extrair as características caóticas do sinal. Em seguida, o servidor persiste esses dados localmente. O protocolo de transporte a ser adotado foi o UDP, visto que, tem grande vantagem na velocidade de comunicação, e a perda mínima de dados não tem grande interferência no sistema.

A análise em tempo real foi implementada como um serviço no servidor. Onde foi definido um intervalo de tempo em que este serviço consulta os dados, e em seguida realiza uma classificação a partir de um modelo de aprendizado de máquina previamente treinado. Detalhes do processo de treinamento e os algoritmos utilizados serão descritos na subseção

de resultados 5.5. A cada classificação realizada, o serviço envia o diagnóstico para uma aplicação web que também foi desenvolvida no trabalho.

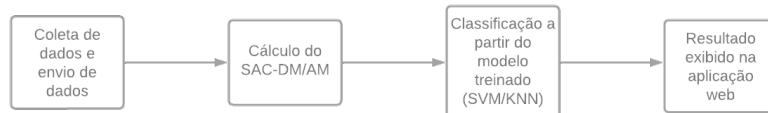
É possível visualizar na figura 4.1, uma ilustração da arquitetura do sistema proposto.

Figura 4.1: Arquitetura do sistema proposto.



A aplicação web foi implementada para facilitar a visualização dos diagnósticos recebidos. Esta aplicação possui dois menus, um para realizar a coleta em um processo automatizado e outro para realizar os diagnósticos. O primeiro menu envia uma requisição ao servidor, que repassa a informação para o dispositivo M5 e este inicia o envio dos sinais de vibração. O servidor começa a receber o sinal, aplica o algoritmo do SAC-DM e realiza a aprendizagem, gerando assim o modelo. O segundo menu também inicia uma operação semelhante ao primeiro, porém o servidor entra no modo de diagnóstico, onde o resultado do SAC-DM é aplicado no modelo gerado e é feito o diagnóstico. A cada diagnóstico, é enviado uma mensagem via MQTT à aplicação, que exibe o resultado para o usuário.

Figura 4.2: Diagrama de blocos do fluxo dos dados.



Na figura 4.2 é possível verificar um diagrama de blocos para facilitar a compreensão do fluxo dos dados. Primeiro é feita a coleta e envio dos dados, em seguida é aplicada uma das técnicas de análise baseada no caos, logo após é realizada a classificação baseada no modelo treinado e por último a apresentação dos diagnósticos na aplicação web.

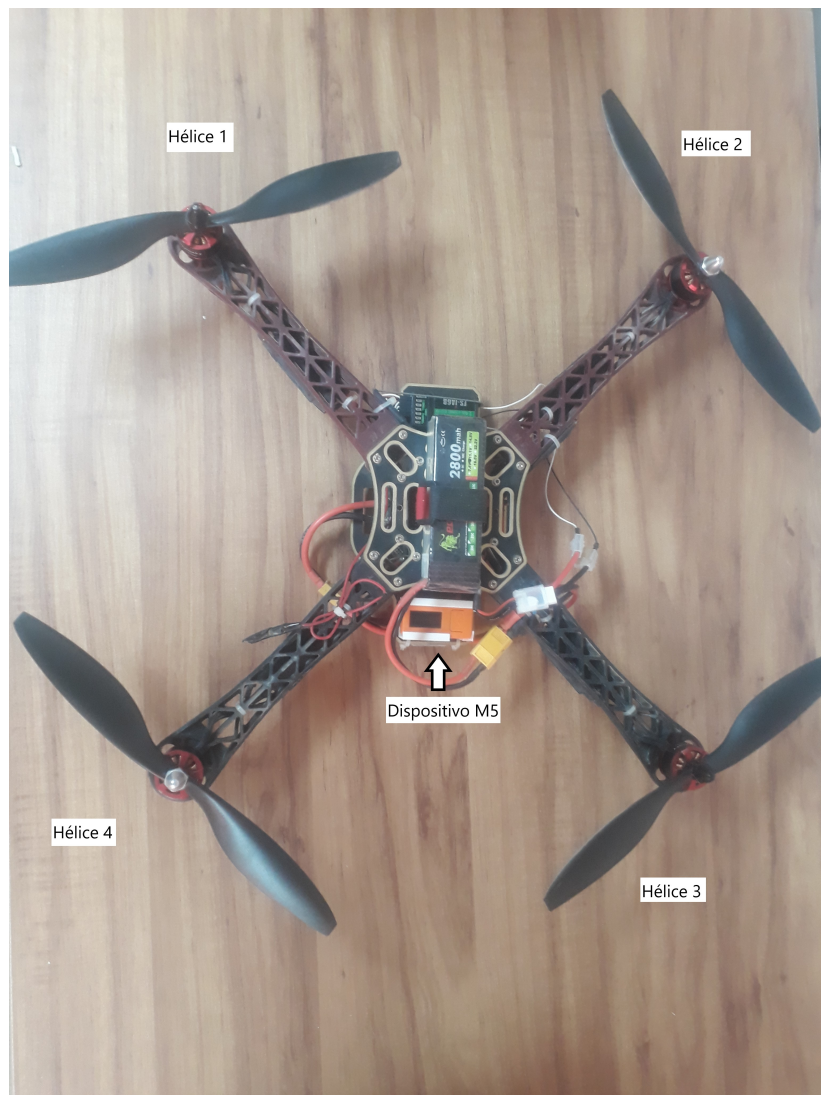
## 4.2 Aquisição do conjunto de dados

O primeiro passo para avaliar a eficácia do método proposto, foi realizar coletadas para formar um conjunto de dados dentro do contexto do objeto de estudo deste trabalho. O conjunto de dados foi obtido através de voos realizados com um drone quadricóptero, coletados em testes controlados. A captura dos dados foi efetuada pelo acelerômetro embarcado ao dispositivo M5, que foi acoplado na carcaça do drone.

O quadricóptero utilizado no experimento possui dimensões de 46 x 46 cm e pode ser visto na figura 4.3. O controlador de voo utilizado foi o APM 2.8, que possui fonte aberta. Além disso, o drone possui Motores 920 Kv, Hélices de tamanho 10, Deslocamento da hélice 45, Peso 1,02 kg, Empuxo máximo 3,44 kg (860 gramas por motor), Transmissor e Receptor (Flysky FS-IA6B) 2A 2.4GHz 6 canais, Controlador de velocidade ESC 30A Simonk e

Bateria 3 células com 2800mAh.

Figura 4.3: Quadricóptero utilizado na aquisição de dados.



A captura dos dados ocorreu num espaço localizado no LASER, que pode ser visto na figura 4.4. Toda operação foi realizada no modo manual, onde o operador manteve um tipo de voo no qual o quadricóptero se encontrou imóvel em relação a um ponto (tipo de voo conhecido na expressão inglesa como *Hover*), logo após atingir uma determinada altitude. O processo se repetiu por 9 voos com duração total de 52 minutos, onde cada voo foi realizado em 6 minutos. Os voos simularam 3 condições físicas do RPA, onde cada condição foi reproduzida em 3 voos. As condições foram: Voo Normal, Condição de Falha 1 e Condição de Falha 2. A tabela 4.1 apresenta detalhes dessas condições:

Figura 4.4: Espaço utilizada para realização dos experimentos.



Tabela 4.1: Cenário dos Voos

Condição	Descrição
Voo Normal (VN)	Voo com o RPA em condições físicas normais.
Condição de Falha 1 (CF1)	Voo com o RPA utilizado uma hélice desbalanceada. (hélice 2, figura 4.3)
Condição de Falha 2 (CF2)	Voo com o RPA utilizado duas hélices desbalanceadas. (hélice 2 e hélice 4, figura 4.3)

Hélices desbalanceadas causam vibrações que podem levar a falhas estruturais mais graves. Dessa forma, é de grande importância manter as hélices do drone balanceadas. Para simular as condições de falhas, foram desbalanceadas duas hélices propositalmente. Uma hélice desbalanceada tende a pender para o lado mais pesado, conforme a figura 4.5 a. Por outro lado, uma hélice balanceada irá se manter estável, conforme figura 4.5 b.

Figura 4.5: Hélice desbalanceada e hélice balanceada.



Foram utilizados no experimento os dados de aceleração de 3 eixos do sensor IMU de 6 eixo (MPU6886), que é integrado ao dispositivo M5. O dispositivo foi posicionado na região de trás do drone, conforme a figura. A taxa de amostragem foi de 500 Hz.

## 4.3 Técnicas utilizadas

### 4.3.1 Implementação da SAC-DM

A implementação de um algoritmo para a SAC-DM, consiste em detectar e contar os picos da amostra, calcular médias de densidade de picos ( $\rho$ ) e calcular o Comprimento de Correlação ( $\text{tal}$ ), conforme a fórmula 3.1. O sinal é dividido em amostras com o mesmo tamanho, onde esse tamanho é definido pelo parâmetro ( $N$ ). A cada iteração do laço de repetição, que irá se repetir por ( $\text{tamanho do sinal/tamanho da amostra}$ ) vezes, o valor do comprimento de correlação é armazenado numa lista denominada SAC-DM, que é retornada pela função. Para ajudar na compreensão é apresentado o algoritmo 1.

---

#### Algorithm 1 SAC-DM

---

```

procedure SAC-DM(data, sampleSize)
   $M \leftarrow \text{size}(\text{data})$ 
   $N \leftarrow \text{sampleSize}$ 
   $\text{start} \leftarrow 0$ 
   $\text{end} \leftarrow N$ 
  while  $k \leq (M/N)$  do
     $\text{peaks} \leftarrow \text{findPeaks}(\text{data}[\text{start} : \text{end}])$ 
     $\text{sacdm}[k] \leftarrow 1.0/6 * (\text{size}(\text{peaks})/N)$ 
     $\text{start} \leftarrow \text{end}$ 
     $\text{end} \leftarrow \text{end} + N$ 
     $k \leftarrow k + 1$ 
  end while
  return sacdm
end procedure

```

---

### 4.3.2 Implementação da SAC-AM

A implementação do SAC-AM é bem similar a do SAC-DM. A diferença é que além de detectar e contar os picos da amostra, é realizado o somatório da amplitude dos máximos e em seguida dividido o resultado pelo tamanho das amostras. O resultado da implementação da técnica pode ser visualizado no algoritmo 2.

---

#### Algorithm 2 SAC-AM

---

```

procedure SAC-AM(data, sampleSize)
   $M \leftarrow \text{size}(\text{data})$ 
   $N \leftarrow \text{sampleSize}$ 
   $\text{start} \leftarrow 0$ 
   $\text{end} \leftarrow N$ 
  while  $k \leq (M/N)$  do
     $\text{peaks} \leftarrow \text{findPeaks}(\text{data}[\text{start} : \text{end}])$ 
     $\text{soma} \leftarrow \text{sum}(\text{peaks})$ 
     $\text{sacam}[k] \leftarrow \text{soma}/N$ 
     $\text{start} \leftarrow \text{end}$ 
     $\text{end} \leftarrow \text{end} + N$ 
     $k \leftarrow k + 1$ 
  end while
  return sacam
end procedure

```

---

### 4.3.3 K-Vizinhos mais Próximos

O k-vizinho mais próximo, do inglês k-Nearest Neighbors (KNN), é um algoritmo baseado na ideia de que amostras mais próximas fornecem informações úteis para previsão do rótulo de um ponto alvo  $x'$ . Com base nos padrões K-mais próximos nas distâncias dos dados, o algoritmo delega os rótulos de classes e dessa forma, é necessário medir a similaridade dos espaços dos dados. A distância Euclidiana é a forma de medida de similaridade amplamente utilizada (ZHANG, 2017).

Outro parâmetro a ser definido no KNN é o tamanho do K-vizinhos, isto é, a quantidade de vizinho que o ponto alvo  $x'$  será comparado. Para um K igual a 1, a previsão se torna local, já para K's maiores o classificador generaliza ignorando pequenos grupos de padrões. Surge então a questão de como escolher o K, que também é conhecido como seleção de modelo. Técnicas como validação cruzada podem ser utilizadas para determinar o parâmetro mais adequado (PEDREGOSA et al., 2011).

#### 4.3.4 Máquinas de Vetor de Suporte

As máquinas de vetores de suporte (SVM) são importantes métodos de aprendizagem supervisionada. Suas aplicações podem ser para classificação, regressão e detecção de outliers (PEDREGOSA et al., 2011). O SVM é um classificador linear binário que passou por ajustes para dados não lineares usando várias técnicas. O modo linear do SVM é considerado bastante eficiente em aplicações de alta dimensão (CHAUHAN; DAHIYA; SHARMA, 2019). Para algumas aplicações o treinamento do kernel linear é mais rápido que o kernel não linear.

### 4.4 Ferramentas utilizadas

A linguagem de programação Python foi utilizada para realizar a análise apresentada neste trabalho. A linguagem foi escolhida por compreender um conjunto de bibliotecas que facilitam os processos fundamentais na ciência de dados. Entre as bibliotecas oferecidas e que foram utilizadas no projeto estão:

- Numpy: Uma biblioteca Python que oferece recursos indispensáveis para a computação científica. Incorpora vários meios de manipular dados com alto desempenho.
- Matplotlib: Biblioteca para visualização de dados e criação de gráficos, utilizada no trabalho para análise visual dos resultados.
- Scikit-learn: Uma biblioteca que oferece recursos para aprendizagem de máquina supervisionada e não supervisionada, bem como recursos para ajuste de modelo, pré-processamento de dados, seleção e avaliação de modelo (PEDREGOSA et al., 2011).

- Scipy: Biblioteca que fornece rotinas para integração numérica, interpolação, otimização, álgebra linear e estatísticas.
- Flask: É um módulo Python que permite desenvolver aplicações web tais como servidores Http.
- Pandas: O pandas é uma ferramenta para análises e manipulação de dados disponível em Python e possui código aberto.

Além dessas ferramentas, também foram utilizados nos trabalhos alguns frameworks, entre eles:

- Ionic: É um kit de ferramentas de interface de usuário, de código aberto, usando tecnologias da web como: HTML, CSS e JavaScript.
- Angular: É um framework que oferece uma estrutura para criação de componentes para aplicações web. Além disso, possui uma série de recursos que permite a integração do aplicativo com servidores.
- MQTT: É um protocolo de mensagens padrão OASIS para a Internet das Coisas.

## 4.5 Avaliação de desempenho

Em análises de classificação é fundamental que se defina as métricas de avaliação de desempenho, para que se possa determinar a qualidade do modelo. Para este trabalho foi escolhido dois métodos de avaliação: Matriz de confusão e acurácia.

### 4.5.1 Matriz de confusão

Matriz de confusão é uma das maneiras mais simples de verificar o desempenho de um modelo de classificação. Nela é possível visualizar a quantidade de classificações corretas e erradas. Ela é formada basicamente pelas quantidades de classificações: Verdadeiros Positivo (VP), Falsos Positivos (FP), Verdadeiros Negativos (VN) e Falsos Negativos (FN). Essas quantidades podem ser expressas em valores absolutos e em porcentagem (SUSMAGA, 2004).

Tabela 4.2: Matriz de confusão

		Preditas	
		Positivo	Negativas
Reais	Positivo	VP	FN
	Negativa	FP	VN

### 4.5.2 Acurácia

A acurácia é outra forma de avaliar o desempenho do modelo, mostrando o quanto de fato foi classificado corretamente. É determinada pela razão entre a quantidade de amostras classificadas corretamente e a quantidade total de amostras.

$$acurcia = \frac{(VP) + (VN)}{(VP) + (VN) + (FP) + (FN)} \quad (4.1)$$

# Capítulo 5

## Resultados

Aqui serão descritos os procedimentos realizados para obtenção dos resultados. Foram realizados dois tipos de experimentos e portanto alcançados dois resultados importantes: O primeiro experimento foi realizado offline, onde foram coletados dados de vibração do drone e posteriormente aplicado algoritmos de classificação, para avaliar a viabilidade da arquitetura proposta em seu funcionamento online. O segundo experimento foi realizado em tempo real (online), baseado em modelos gerados a partir dos dados obtidos no primeiro experimento. A sequência de atividades para geração dos resultados foram: Configuração dos teste, Criação do Software Embarcado no Dispositivo M5, Implementação do Servidor e do Serviço de Análise, Implementação do Aplicativo Web, Teste e Validação do sistema (*off-line*) e Teste e Validação do sistema (*on-line*).

### 5.1 Configuração dos teste

Nesta etapa foram realizados ajustes no drone para a realização dos experimentos e implantação das falhas objeto de identificação. As falhas de desbalanceamento de hélices, descritas no capítulo 5, foram implantadas no drone utilizando fita isolante com um tamanho de 5 cm. As fitas foram coladas na parte inferior das hélices 2 e 4, de acordo com a condição de falha definida na tabela 4.1. Além disso foi feita a calibração dos sensores presentes na controladora de voo do drone, para garantir uma uniformidade na pilotagem.

## 5.2 Programação Sistema Embarcado

A implementação do software embarcado no dispositivo M5, foi realizada na linguagem C++, utilizando a plataforma Arduino. O sistema desenvolvido foi capaz de capturar os dados do sensor IMU integrado ao dispositivo e transmitir os dados via websocket. Além disso, o sistema possibilitou ao dispositivo conectar-se a um broker MQTT para receber o comando de iniciar e parar as coletas.

## 5.3 Implementação do Servidor e do Serviço de Análise

O servidor foi implementado utilizando o framework Flask, utilizando a linguagem Python. O servidor desenvolvido utiliza os padrões Rest para atender requisições da Aplicação Web e do dispositivo M5. O servidor atende a esses clientes através de requisições com métodos GET e POST nos padrões HTTP e por conexões websocket via UDP. As funcionalidades do servidor são:

- Atender requisições da aplicação web, que são: Iniciar coleta e realizar diagnóstico;
- Enviar comando via websocket para o dispositivo M5 iniciar o envio dos dados de vibração;
- Receber via websocket os sinais enviados pelo dispositivo;
- Fazer o armazenamento dos dados recebidos em arquivos.

O servidor também inclui o serviço de análise online. Uma vez gerados os modelos dos dados nos classificadores, o serviço utiliza-os como base para realizar novas predições e enviar o resultado do diagnóstico para a aplicação web via MQTT.

## 5.4 Implementação do Aplicativo Web

A aplicação web foi desenvolvida com o objetivo de automatizar as coletas para validação offline e facilitar a visualização dos diagnósticos na validação online. A aplicação foi desenvolvida utilizando o framework Angular, que engloba as linguagens web como: HTML, CSS

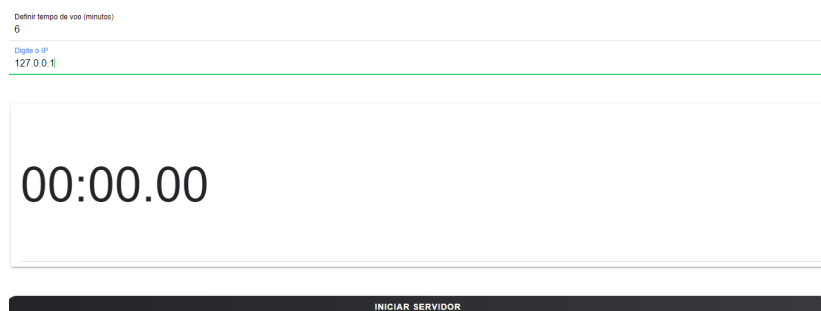
e JavaScript. Também foi utilizada a biblioteca de componentes de interface visual Ionic. A aplicação possui uma tela inicial, como pode ser vista na figura 5.1, que apresenta suas duas principais funcionalidades: Realizar coletas e realizar diagnóstico.

Figura 5.1: Tela inicial da aplicação.



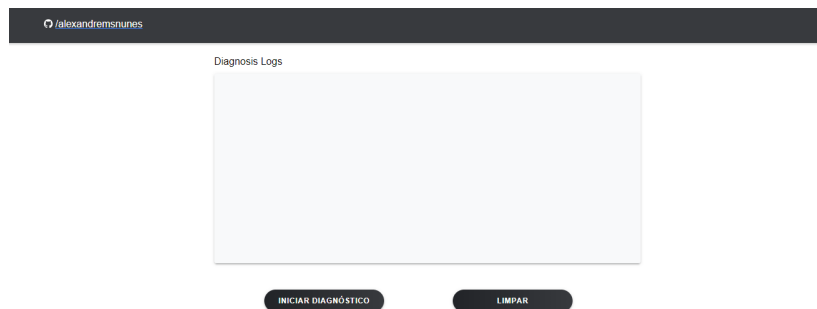
Na tela de realizar coleta são exibidos dois campos: 'Definir tempo de voo' e 'Digite o IP'. O primeiro campo configura o tempo de voo do experimento, o segundo define o endereço IP do dispositivo M5. Logo abaixo dos campos é exibido um cronômetro para o operador acompanhar a contagem do tempo que foi definida. Em seguida é apresentado um botão 'Iniciar Servidor' para iniciar o experimento, conforme figura 5.2.

Figura 5.2: Tela para realizar coletas.



Quando o botão iniciar é pressionado a aplicação envia uma requisição HTTP como o método POST enviando os parâmetros definidos. Ao receber, o servidor envia uma mensagem MQTT para o dispositivo M5, que começa a enviar os pacotes de dados. Após o término do tempo definido, o servidor envia uma mensagem ao M5 solicitando o encerramento do envio, salva os dados em um arquivo e aguarda novas requisições da aplicação.

Figura 5.3: Tela para realizar diagnósticos.



Na tela de realizar diagnóstico, que pode ser vista na figura 5.3, é apresentado um componente visual chamado de card, no qual assume a função de apresentar os Logs de diagnósticos. Abaixo do card existem dois botões: ‘Iniciar diagnóstico’ e ‘Limpar’. O primeiro botão é responsável por enviar uma requisição HTTP para o servidor solicitando o início do modo diagnóstico. O servidor envia uma mensagem para o dispositivo solicitando envio dos sinais e realiza as predições baseadas no modelo gerado. Os diagnósticos são apresentados no Log a cada 30 segundos.

## 5.5 Teste e Validação do sistema (off-line)

Após a aquisição dos dados descrita no capítulo 5, foi realizada a junção da coleta de cada voo para o respectivo cenário. A partir disso, foi iniciada a análise offline, para avaliar o desempenho das técnicas e geração do modelo a ser utilizado na etapa online. Esta análise foi dividida em três etapas: análise com os dados brutos, análise com a aplicação da técnica SAC-DM e análise com a aplicação da SAC-AM. O objetivo desta divisão foi avaliar o quanto cada técnica pode melhorar individualmente os resultados de classificação e, dessa forma, se mostrar relevante para o propósito de diagnosticar falhas em RPA’s.

A avaliação foi produzida em classificações entre os dados dos três cenários propostos: Voo Normal (VN), Condição de Falha (CF1) e Condição de Falha (CF2). Os dados aplicados na entrada são os três eixos do acelerômetro. Antes da realização do treinamento dos algoritmos de classificação, foi realizado um balanceamento nos dados para suprir as diferenças nos tamanhos dos conjuntos de cada cenário. Em seguida foi procedido uma divisão nos

dados formando um conjunto para testes de 30% e outro conjunto para treinamento de 70%, a fim de evitar problemas de sobre-ajustes. Também foi realizada a aplicação da técnica de validação cruzada para avaliar a capacidade de generalização do modelo. São apresentados nas figuras 5.4, 5.5 e 5.6 um comparativo entre os cenários dos sinais brutos, com SAC-DM aplicado e SAC-AM aplicado, respectivamente.

Figura 5.4: Comparativo entre os sinais brutos do voo normal, voo com falha 2 e voo com falha 3, para os três eixos do acelerômetro.

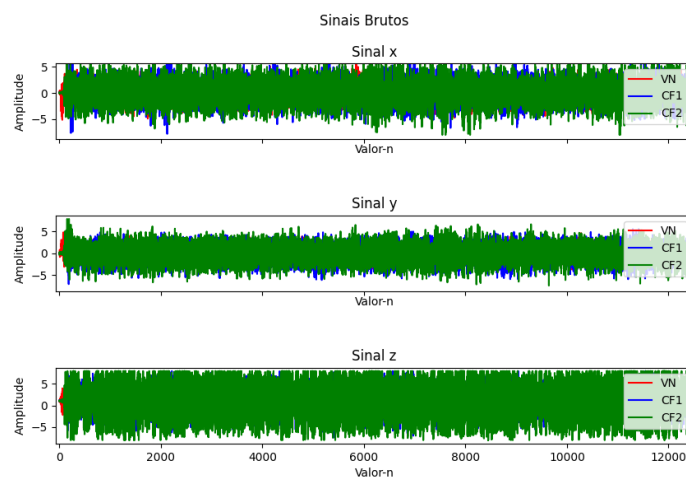


Figura 5.5: Comparativo entre os sinais com SACDM do voo normal, voo com falha 2 e voo com falha 3, para os três eixos do acelerômetro.

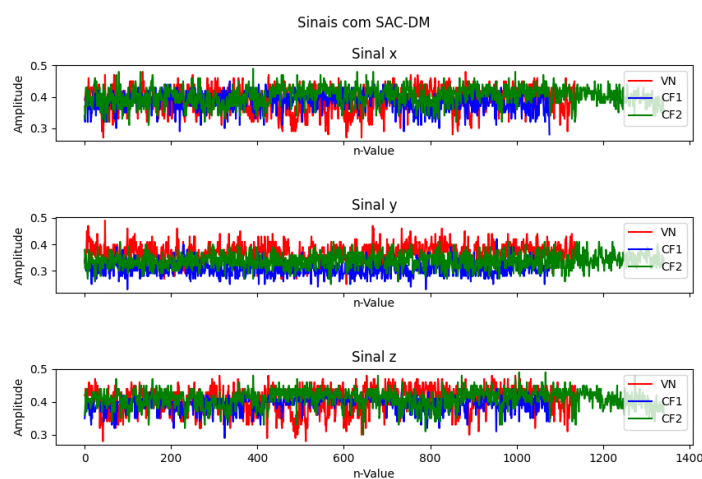
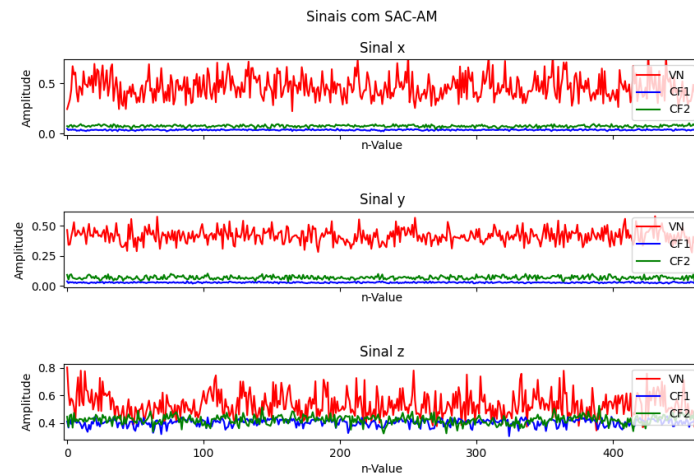


Figura 5.6: Comparativo entre os sinais com SACAM do voo normal, voo com falha 2 e voo com falha 3, para os três eixos do acelerômetro.



Uma vez separados o conjunto de treinamento do conjunto de testes, a análise avançou para os treinamentos dos classificadores. Para o classificador KNN foi definido como 3 o número de vizinhos a serem usados para consultas. Já no classificador SVM, foi adotado o valor experimentado para os parâmetros que apresentassem o melhor resultado. Nas figuras 5.7, 5.8 e 5.9, são apresentados a matriz de confusão normalizadas com os resultados de diagnóstico do algoritmo KNN, para os sinais brutos, com SAC-DM aplicado e SAC-AM aplicado, respectivamente.

Figura 5.7: Desempenho do algoritmo KNN para os sinais brutos.

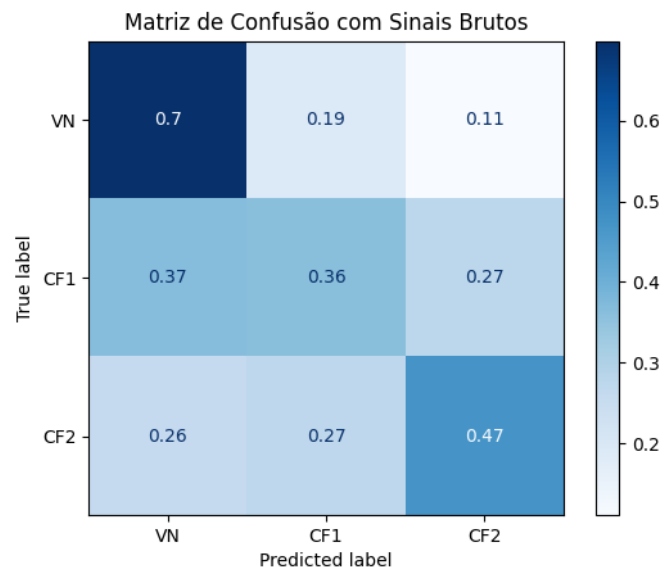


Figura 5.8: Desempenho do algoritmo KNN para o SAC-DM aplicado.

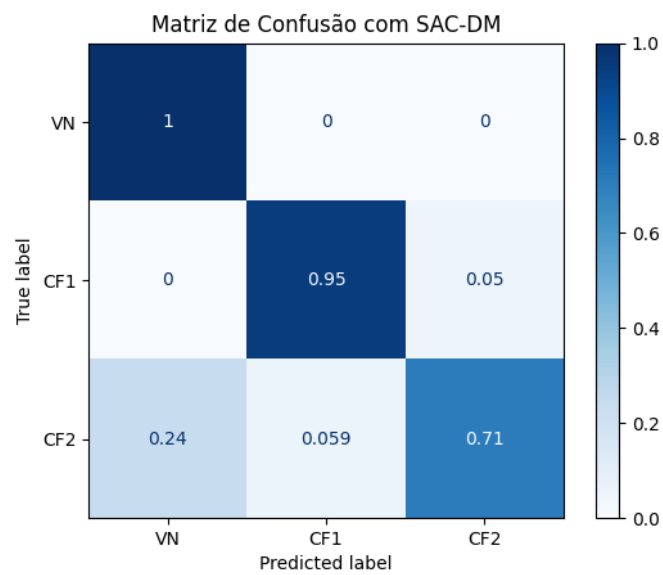
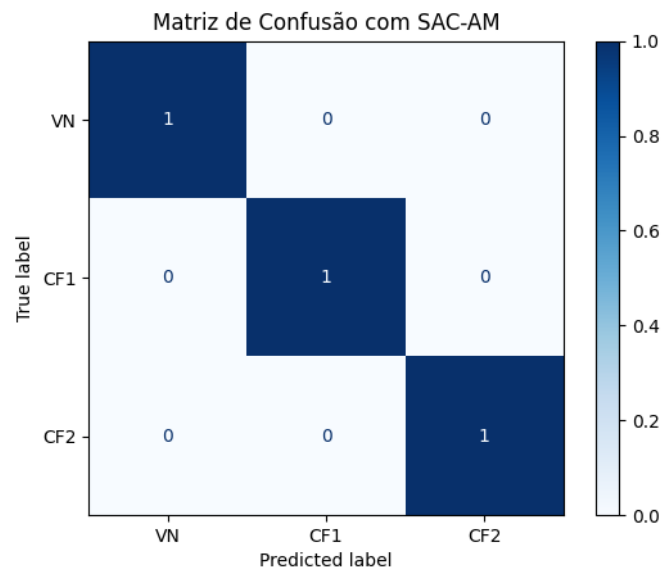


Figura 5.9: Desempenho do algoritmo KNN para o SAC-AM aplicado.



O desempenho apresentado para o algoritmo KNN para os sinais brutos foi de 51% de acurácia. Já com o SAC-DM aplicado o melhor resultado para o algoritmo foi de 91%. O melhor resultado do SAC-DM foi obtido testando diferentes tamanho das amostras, iniciando com tamanho 100. O melhor valor para o tamanho das amostras, isto é, o valor para calcular um resultado do SAC-DM, foi de 1000. Já para o SAC-AM a acurácia foi de 100% para um tamanho de amostras com valor 100. Com isso, não houve a necessidade de aumentar o tamanho das para o SAC-AM.

Figura 5.10: Desempenho do algoritmo SVM para os sinais brutos.

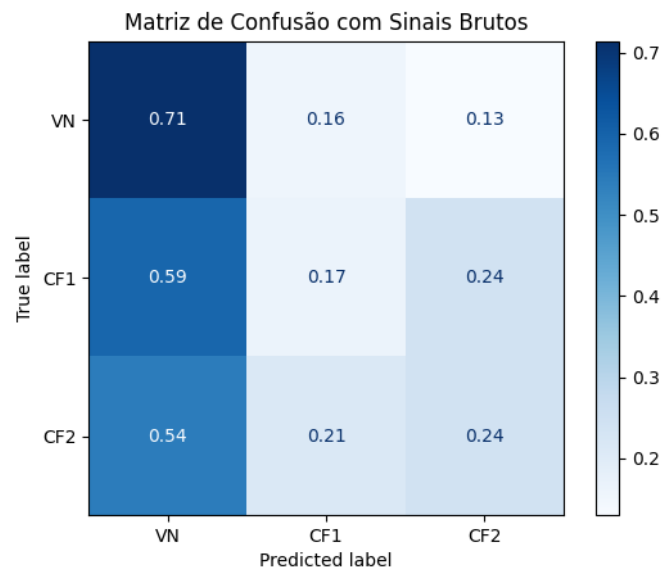


Figura 5.11: Desempenho do algoritmo SVM para o SAC-DM aplicado.

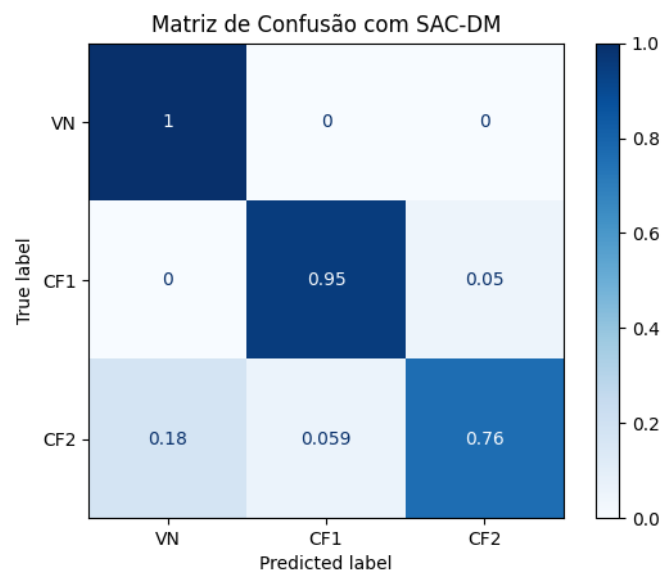
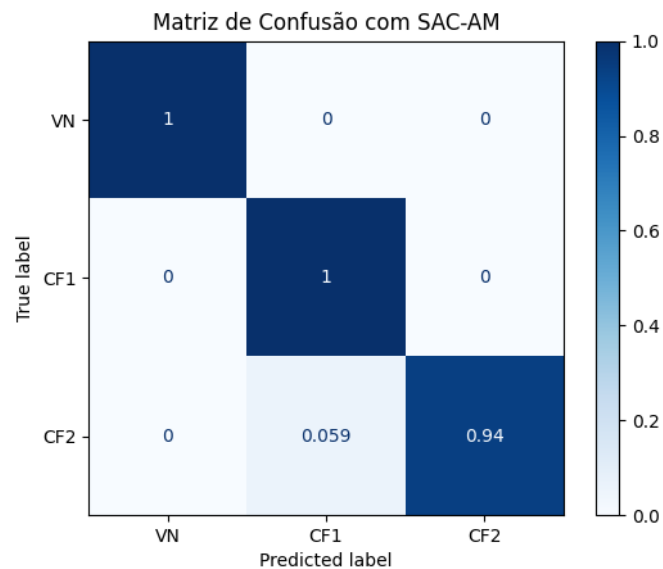


Figura 5.12: Desempenho do algoritmo SVM para o SAC-AM aplicado.



As figuras 5.10, 5.11 e 5.12 apresentam os resultados para o algoritmo SVM. Para os sinais brutos, o algoritmo obteve desempenho de 37%. Já para o SAC-DM também foi necessário testar diferentes tamanhos de amostras. O tamanho com o melhor resultado também foi de 1000 apresentando uma acurácia de 90%. Para o SAC-AM o desempenho foi de 98% para 100 amostras. Na tabela 5.1 é possível visualizar o desempenho de cada algoritmo para cada técnica.

Tabela 5.1: Desempenhos dos algoritmos para cada técnica.

	KNN	SVM
Sinais Brutos	51%	37%
SAC-DM Aplicado	91%	90%
SAC-AM Aplicado	100%	98%

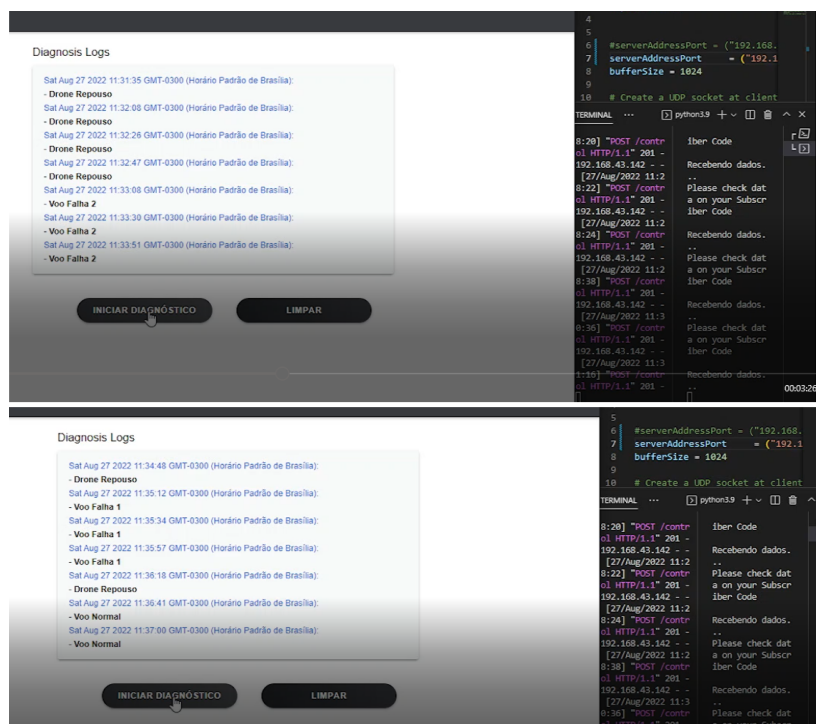
Pode ser notada uma melhora significativa no desempenho para ambas as técnicas, comparadas aos sinais brutos. Os resultados obtidos no experimento *off-line* geraram o modelo a ser utilizado no experimento *on-line*. O algoritmo KNN levou uma certa vantagem no desempenho apresentado, dessa forma foi escolhido gerar o modelo baseado neste algoritmo.

## 5.6 Teste e Validação do sistema (on-line)

Na etapa on-line a análise foi realizada ao mesmo tempo que a captura, ou seja, em tempo real, a partir do modelo gerado na análise off-line. O experimento foi realizado de acordo com o seguinte fluxo:

1. O drone iniciou com as duas fitas nas hélices 2 e 4, configurando a condição de falha 2.
2. Após algumas predições corretas, o drone foi pousado e foi retirada a fita da hélice 4. Restando apenas a fita na hélice 2, caracterizando a condição de falha 1. Após isso, foi levantado voo novamente.
3. Em seguida, após algumas predições, o drone foi novamente pousado e retirado a fita da hélice 2. Caracterizando assim um voo normal.

Figura 5.13: Tela de diagnósticos apresentando alguns resultados



Neste experimento o algoritmo utilizado foi o KNN e a técnica utilizada foi a SAC-AM. Para essa configuração, em todos os cenários, o sistema foi capaz de identificar corretamente

a condição que foi configurada. Vale também ressaltar que foi adicionada mais uma condição chamada ‘Drone Repouso’, para identificar o momento em que o drone encontrava-se ao solo. Para isso, foi capturado dados do drone em repouso pela mesma quantidade tempo dos outros cenários, em seguida foi gerado um novo modelo com a nova condição. Na figura 5.13 podemos ver o log da tela de diagnósticos apresentando alguns resultados do experimento. A imagem foi retirada de um vídeo gravado do experimento.

Tabela 5.2: Desempenhos final do experimento on-line.

	Acertos	Erros
Drone Repouso	100%	0%
Voo Normal ( <b>Hover</b> )	100%	0%
Voo Com Falha 1 ( <b>Hover</b> )	100%	0%
Voo Com Falha 2 ( <b>Hover</b> )	100%	0%

Na tabela 5.2 é possível ver o resultado final de cada cenário em tempo real. Para todos os cenários o sistema acertou 100% das vezes, demonstrando sua boa eficiência apresentada na tabela 5.1.

# Capítulo 6

## Conclusão

Este trabalho propôs uma arquitetura de um sistema para realização de falhas em drones, em tempo real. A aplicação das técnicas SAC-DM e SAC-AM demonstraram um ganho significativo no desempenho dos diagnósticos das falhas. Nos processos tanto off-line quanto on-line constataram que a aplicação da arquitetura proposta é relativamente simples e o custo de implantação pode ser considerado baixo, demonstrando boa viabilidade técnica e financeira. Dessa forma, os objetivos geral e específicos foram completamente alcançados.

O sistema apresentado é composto por um RPA quadricóptero, um dispositivo que foi capaz de capturar dados de vibração do drone e realizar a transmissão dos dados, uma aplicação web capaz de solicitar o envio de dados e apresentar os resultados dos diagnósticos, um servidor capaz de atender às requisições do dispositivo embarcado e aplicação web, bem como aplicar aprendizagem de máquina para realizar os diagnósticos.

Diante dos bons resultados apresentados na pesquisa, foi possibilitado a submissão e publicação do artigo “Failure Detection and Diagnosis of a Small Unmanned Aerial Vehicle using Amplitude of Peaks” na vigésima sexta conferência "IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)".

Conclui-se que o trabalho atingiu o objetivo geral, onde se conseguiu demonstrar que a arquitetura do sistema de diagnóstico de falhas proposta foi capaz de ser utilizada em tempo real e apresentando bom desempenho, o que permite auxiliar na definição de estratégias de segurança do voo. Também foram alcançados os objetivos específicos de aplicar os métodos SAC-AM e SAC-DM e demonstrar a viabilidade técnica e financeira do projeto. Diante disso, é possível indicar que o trabalho contribuiu com a eficácia e manutenção de RPAs, po-

dendo contribuir com a melhoria dos produtos, tornando-os mais seguros para os operadores e pessoas no ambiente.

Como sugestões para trabalhos futuros pode-se citar: Desenvolvimento bibliográfico da Análise de Amplitude de Máximos e Evolução do sistema para falhas de comunicação, nos sensores e fatores ambientais.

## Referências Bibliográficas

- [1] ANAC. Agência Nacional De Aviação Civil. 2020. Disponível em: <<https://www.anac.gov.br/assuntos/paginas-tematicas/drones>>. Acesso em: 20 de Abr. de 2020.
- [2] ANGULAR. Disponível em: <<https://angular.io/>>. Acesso em: 10 de Mar. de 2021.
- [3] BASKAYA, Elgiz; BRONZ, Murat; DELAHAYE, Daniel. Fault detection diagnosis for small UAVs via machine learning. In: 2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC). IEEE, 2017. p. 1-6.
- [4] BEKTASH, O. M.; LA COUR-HARBO, Anders. Vibration Analysis for Anomaly Detection in Unmanned Aircraft. In: ANNUAL CONFERENCE OF THE PROGNOSTICS AND HEALTH MANAGEMENT SOCIETY 2020. PHM Society, 2020.
- [5] BONACCORSO, Giuseppe. Machine learning algorithms. Packt Publishing Ltd, 2017.
- [6] BONDYRA, A. et al. Fault diagnosis and condition monitoring of uav rotor using signal processing. In: 2017 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA). IEEE, 2017. p. 233-238.
- [7] CAMBEL, A. B.; Applied chaos theory: A paradigm for complexity. Elsevier, 1993.
- [8] CHAUHAN, Vinod Kumar; DAHIYA, Kalpana; SHARMA, Anuj. Problem formulations and solvers in linear SVM: a review. Artificial Intelligence Review, v. 52, n. 2, p. 803-855, 2019.
- [9] COOK, Kendra LB. The silent force multiplier: The history and role of UAVs in warfare. In: 2007 IEEE Aerospace Conference. IEEE, 2007. p. 1-7.

- [10] CORK, Lennon; WALKER, Rodney; DUNN, Shane. Fault detection, identification and accommodation techniques for unmanned airborne vehicles. In: Proceedings Australian international aerospace congress. AIAC, 2005. p. 1-18.
- [11] DALAMAGKIDIS, Konstantinos; VALAVANIS, Kimon P.; PIEGL, Les A. (Ed.). On integrating unmanned aircraft systems into the national airspace system: issues, challenges, operational restrictions, certification, and recommendations. Dordrecht: Springer Netherlands, 2009.
- [12] DECEA. Departamento de Controle do Espaço Aéreo. 2020. Disponível em: <<https://www.anac.gov.br/assuntos/paginas-tematicas/drones>>. Acesso em: 20 de Abr. de 2020.
- [13] DITTO, William; MUNAKATA, Toshinori. Principles and applications of chaotic systems. *Communications of the ACM*, v. 38, n. 11, p. 96-102, 1995.
- [14] EL NAQA, Issam; MURPHY, Martin J. What is machine learning?. In: *machine learning in radiation oncology*. Springer, Cham, 2015. p. 3-11.
- [15] FLASK. Disponível em: <<https://flask.palletsprojects.com/>>. Acesso em: 10 de Mar. de 2021.
- [16] FREEMAN, Paul, et al. "Model-based and data-driven fault detection performance for a small UAV." *IEEE/ASME Transactions on mechatronics* 18.4 (2013): 1300-1309.
- [17] GHALAMCHI, Behnam; JIA, Zheng; MUELLER, Mark Wilfried. Real-time vibration-based propeller fault diagnosis for multicopters. *IEEE/ASME Transactions on Mechatronics*, v. 25, n. 1, p. 395-405, 2019.
- [18] HONG, Jun-Ki; LEE, Yang-Kyoo. Deep Learning based Abnormal Vibration Prediction of Drone. *Journal of Internet Computing and Services*, v. 22, n. 3, p. 67-73, 2021.
- [19] ICAO. International Civil Aviation Organization. 2020. Disponível em: <[www.icao.int](http://www.icao.int)>. Acesso em: 20 de Abr. de 2020.
- [20] IONIC. Disponível em: <<https://ionicframework.com/>>. Acesso em: 10 de Mar. de 2021.

- [21] JAZAYERI, Mehdi. Some trends in web application development. In: Future of Software Engineering (FOSE'07). IEEE, 2007. p. 199-213.
- [22] KING, David W.; BERTAPELLE, Allen; MOSES, Chad. UAV failure rate criteria for equivalent level of safety. In: International helicopter safety symposium. 2005.
- [23] LAKSHMANAN, M.; RAJASEKAR, S. Chaos in dissipative nonlinear oscillators and criteria for chaos. In: Nonlinear Dynamics. Springer, Berlin, Heidelberg, 2003. p. 123-158.
- [24] LÓPEZ-ESTRADA, Francisco-Ronay et al. Detección de fallas en vehículos aéreos no tripulados mediante señales de orientación y técnicas de aprendizaje de máquina. Revista Iberoamericana de Automática e Informática industrial, 2021.
- [25] MAGRI, João Alexandre. Criando e usando web service. Augusto Guzzo Revista Acadêmica, n. 11, p. 166-183, 2013.
- [26] MAHESH, Batta. Machine learning algorithms-a review. International Journal of Science and Research (IJSR).[Internet], v. 9, p. 381-386, 2020.
- [27] MATPLOTLIB. Disponível em: <<https://matplotlib.org/>>. Acesso em: 10 de Mar. de 2021.
- [28] MEDEIROS, Ramon Leonn Victor et al. Método de detecção de falhas em motores decorrente contínua sem escovas utilizando análise do caos. 2018. Tese (Doutorado) - Programa de Pós-Graduação em Engenharia Mecânica, Centro de Tecnologia, Universidade Federal da Paraíba, João Pessoa. 2018.
- [29] MORALES, A. C.; PAEZ, D.; ARANGO, C. Multi-criteria analysis of UAVs regulations in 6 countries using the analytical hierarchical process and expert knowledge. The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, v. 40, n. 1, p. 175, 2015.
- [30] MQTT. Disponível em: <<https://mqtt.org/>>. Acesso em: 10 de Mar. de 2021.
- [31] M5STACK. Kit de Ferramentas de Desenvolvimento de Código Aberto. 2020. Disponível em: <<https://m5stack.com/>>. Acesso em: 25 de Abr. de 2020.

- [32] NUMPY. Disponível em: <<https://numpy.org/>>. Acesso em: 10 de Mar. de 2021.
- [33] PANDAS. Disponível em: <<https://pandas.pydata.org/>>. Acesso em: 10 de Mar. de 2021.
- [34] PEDOTTI, Luciane Agnoletti dos Santos et al. Dispositivo iot de baixo custo para diagnóstico de falhas em máquinas rotativas. 2019. Tese (Doutorado) - Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas. 2019.
- [35] PEDREGOSA F. et al., “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011
- [36] PRISACARIU, Vasile et al. The history and the evolution of UAVs from the beginning till the 70s. *Journal of Defense Resources Management (JoDRM)*, v. 8, n. 1, p. 181-189, 2017.
- [37] SANTOS, Matheus. (2017). Veículos aéreos não tripulados e sua inserção em espaço aéreo não segregado. *Revista Vértices*. 18. 173. 10.19180/1809-2667.v18n32016p173-188.
- [38] SAMUEL, Arthur L. Machine learning. *The Technology Review*, v. 62, n. 1, p. 42-45, 1959.
- [39] SCPY. Disponível em: <<https://scipy.org/>>. Acesso em: 10 de Mar. de 2021.
- [40] SHINDE, Pramila P.; SHAH, Seema. A review of machine learning and deep learning applications. In: 2018 Fourth international conference on computing communication control and automation (ICCUBEA). IEEE, 2018. p. 1-6.
- [41] SHIMIZU, Kuniyasu; SEKIKAWA, Munehisa; INABA, Naohiko. Mixed-mode oscillations and chaos from a simple second-order oscillator under weak periodic perturbation. *Physics Letters A*, v. 375, n. 14, p. 1566-1569, 2011.
- [42] SOUZA, Jefferson S. et al. Motor speed estimation and failure detection of a small UAV using density of maxima. *Frontiers of Information Technology Electronic Engineering*, v. 22, n. 7, p. 1002-1009, 2021.

- [43] SUSMAGA, Robert. Confusion matrix visualization. In: Intelligent information processing and web mining. Springer, Berlin, Heidelberg, 2004. p. 107-116.
- [44] TANENBAUM, Andrew S. Redes de Computadores. ed. Campus-Tradução da Terceira Edição, Rio de Janeiro, 2003.
- [45] TIHOMIROVS, Juris; GRABIS, Jānis. Comparison of soap and rest based web services using software evaluation metrics. Information technology and management science, v. 19, n. 1, p. 92-97, 2016.
- [46] VÉRAS, Frank César Lopes et al. Detecção de velocidade e de falha de excentricidade em motores elétricos a partir de sinais sonoros utilizando densidade de máximos. 2019. Tese (Doutorado) - Programa de Pós-Graduação em Engenharia Mecânica, Centro de Tecnologia, Universidade Federal da Paraíba, João Pessoa. 2019.
- [47] WILLIAMS, Garnett. Chaos theory tamed. CRC Press, 1997.
- [48] ZHANG, X. et al. Fault detection and identification method for quadcopter based on airframe vibration signals. Sensors, v. 21, n. 2, p. 581, 2021.
- [49] ZHANG, Shichao et al. Learning k for knn classification. ACM Transactions on Intelligent Systems and Technology (TIST), v. 8, n. 3, p. 1-19, 2017.
- [50] ZHONG, Wang; XIN, Chen. Fault detection of UAV fault based on a SFUKF. In: IOP Conference Series: Materials Science and Engineering. IOP Publishing, 2019. p. 052099.