



Universidade de Brasília
Faculdade de Tecnologia

**Identificação de Falhas em Sistemas Reais
e suas Implementações em Dispositivos
Embarcados**

Gabriel Reves Vasques Tonussi

DISSERTAÇÃO DE MESTRADO
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Brasília
2024

Universidade de Brasília
Faculdade de Tecnologia

**Identificação de Falhas em Sistemas Reais
e suas Implementações em Dispositivos
Embarcados**

Gabriel Reves Vasques Tonussi

Documento apresentado como requisito para
conclusão do programa de mestrado em Enge-
nharia Mecatrônica

Orientador (in memoriam): Prof. Dr. Carlos Humberto Llanos
Coorientador: Prof. Dr. Daniel Mauricio Muñoz

Brasília
2024

FICHA CATALOGRÁFICA

Reves Vasques Tonussi, Gabriel.

Identificação de Falhas em Sistemas Reais e suas Implementações em Dispositivos Embarcados / Gabriel Reves Vasques Tonussi; orientador (in memoriam) Carlos Humberto Llanos; coorientador Daniel Mauricio Muñoz. -- Brasília, 2024.

86 p.

Dissertação de Mestrado (Engenharia de Controle e Automação)
-- Universidade de Brasília, 2024.

1. Identificação de Falhas. 2. Identificação de Sistemas. 3. Respirador Mecânico. 4. Aprendizado de Máquinas. 5. Sistemas Embarcados.I. Llanos, Carlos Humberto, orient. II. Muñoz, Daniel Mauricio, coorient. III. Título.

**Universidade de Brasília
Faculdade de Tecnologia**

**Identificação de Falhas em Sistemas Reais e suas
Implementações em Dispositivos Embarcados**

Gabriel Reves Vasques Tonussi

Documento apresentado como requisito para
conclusão do programa de mestrado em Enge-
nharia Mecatrônica

Trabalho aprovado. Brasília, 7 de dezembro de 2024:

Prof. Dr. Daniel Mauricio Muñoz Arboleda,
UnB
Co-Orientador

Prof. Dr. Fernando Buarque de Lima Neto,
UPE
Examinador externo

Prof. Dr. Sanderson César Macedo Barbalho
Examinador interno

Brasília
2024

Ao meu Orientador, Prof. Dr. Carlos Llanos (in memoriam)

Agradecimentos

Agradeço em primeiro lugar à Universidade de Brasília e seus professores, em especial o Prof. Dr. Carlos Humberto Llanos por sua orientação nas minhas pesquisas e trabalhos científicos. Agradeço também à equipe do LEIA e ao Professor Dr. Daniel Muñoz, por assumir a co-orientação do meu trabalho.

Agradeço a FAPDF, pela bolsa de pesquisa oferecida para o desenvolvimento do projeto *Desenvolvimento de respirador mecânico de baixo custo com sistemas de controle de volume e pressão e adequado às condições sanitárias para pacientes em UTI devido ao COVID-19*.

Agradeço também ao CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brazil) pela bolsa de pesquisa oferecida ao longo do meu mestrado, enquanto não tive vínculo empregatício.

Agradeço às pessoas que me ajudaram e apoiaram ao longo do desenvolvimento dessa tese:

À minha namorada Milene.

À minha família.

Aos meus amigos de faculdade.

Aos amigos que conheci trabalhando na Embraer e na Boeing.

Resumo

Embora exista uma variedade de técnicas e algoritmos na área de identificação de falhas, a viabilidade de suas implementações em aplicações embarcadas reais ainda apresenta algumas lacunas a serem pesquisadas. Neste trabalho, serão explorados três estudos de caso diferentes que apresentaram resultados práticos e de valor científico. O primeiro estudo de caso aborda uma arquitetura aplicada ao monitoramento em tempo real de falhas em materiais. O segundo, uma arquitetura voltada para a detecção de falhas em dutos. Por fim, o terceiro se concentra na detecção de falhas no sistema de sensoriamento de um respirador mecânico de baixo custo desenvolvido pela UnB.

Através do uso de técnicas de sensoriamento ultrassônico baseadas no efeito acustoeelástico, é possível estimar potenciais falhas em materiais, avaliando o estresse e as deformações a que estão submetidos. O sensoriamento ultrassônico também pode ser aplicado na identificação de falhas em soldas, permitindo o mapeamento de possíveis trincas e falhas mecânicas em pontos críticos por meio da análise de imagens que revelam descontinuidades inesperadas. Neste trabalho, foi explorada a otimização das técnicas de alto custo computacional normalmente empregadas para o processamento desses dados ultrassônicos. Selecionando assim, a identificação de falhas em materiais como primeiro estudo de caso e a identificação de falhas em soldas como segundo estudo de caso.

O algoritmo selecionado para a detecção de falhas nos dois primeiros estudos de caso foi uma combinação do *Principal Component Analysis* (PCA), utilizado para a redução da dimensionalidade dos dados, e do regressor *Least Generalized Regression* (LGR), responsável por determinar a presença ou ausência de falhas. A decisão de utilizar o PCA como ferramenta de redução de dimensionalidade foi feita por conta de sua simplicidade de implementação e fácil paralelização, visto que a seção de maior custo computacional do algoritmo se resume a um simples multiplicação matricial. Já a escolha do LGR foi motivada pelo seu baixo custo computacional na fase de inferência, permitindo a rápida execução do algoritmo após a determinação dos parâmetros do regressor.

No contexto dos dois primeiros estudos de caso, uma nova arquitetura é apresentada para otimizar o problema de detecção de falhas em tempo real de materiais e soldas através da utilização de algoritmos de monitoramento de alto custo computacional. Uma vez que os parâmetros que representam o sistema sem falhas são carregados na arquitetura, o processamento de estímulos e respostas ultrassônicas permite a detecção de possíveis falhas no problema monitorado, através de uma análise de regressão do resultado do LGR.

A arquitetura proposta apresenta resultados concretos de aceleração do desempenho do algoritmo PCA+LGR, apresentando um ganho de performance de pelo menos 12 vezes em relação a implementação do algoritmo em um processador ARM comercial. A arquitetura

apresentada utiliza um dispositivo SoC (*System-on-Chip*), que integra *hardware* (FPGA) e *software* (ARM), para a implementação do modelo de identificação, visando maximizar o desempenho computacional e minimizar o consumo de energia. Os resultados finais de validação demonstraram alta precisão em um tempo significativamente menor do que o utilizado por um microcontrolador convencional, mesmo empregando uma quantidade reduzida de recursos computacionais do chip SoC-FPGA.

Outra maneira de identificar falhas pode ser feita através da comparação entre a saída de uma planta estimada e as saídas da planta real. Essa estratégia é apresentada no terceiro estudo de caso, que detecta possíveis falhas em um sistema de sensoriamento de pressão integrado a um respirador mecânico real. Para estimar a planta do respirador mecânico, foi utilizada uma rede neural *Radial Basis Function* (LGR), que através de uma série de experimentos, foi capaz de estimar a pressão na linha principal do respirador através do uso dos dados de fluxo de ar e concentração de oxigênio.

Para avaliar a efetividade do modelo de identificação de falhas no terceiro estudo de caso, foi realizada uma validação da planta estimada com base no critério estatístico *R-squared* (R²), aplicado às saídas do modelo operando em *Free-Run* (FR) e *One-Step-Ahead* (OSA). Para esse caso, a planta identificada apresentou um critério R² de 0.66 para FR e 0.82 para OSA. Apesar dos valores de R² não terem sido altos, eles foram suficientes para a aplicação proposta, visto que o objetivo do trabalho foi realizar uma comparação simples entre o valor estimado e o valor real para a identificação de falhas e descontinuidades no sistema de sensoriamento de pressão.

Palavras-chave: Identificação de Falhas. Identificação de Sistemas. Respirador Mecânico. Aprendizado de Máquinas. Sistemas Embarcados.

Abstract

Although there are various techniques and algorithms in the field of fault identification, the feasibility of their implementation in real embedded applications still presents some gaps to be explored. In this work, three different case studies that yielded practical and scientifically valuable results are explored. The first case study addresses an architecture applied to real-time monitoring of material failures. The second focuses on an architecture for fault detection in pipelines. Finally, the third case study is centered on the detection of faults in the sensing system of a low-cost mechanical ventilator developed by UnB.

Through the use of ultrasonic sensing techniques based on the acoustoelastic effect, it is possible to estimate potential material failures by evaluating the stress and deformations to which they are subjected. Ultrasonic sensing can also be applied to the identification of welding failures, allowing the mapping of potential cracks and mechanical faults at critical points through image analysis that reveals unexpected discontinuities. In this work, the optimization of computationally expensive techniques commonly used for processing these ultrasonic data was explored. Thus, fault identification in materials was selected as the first case study, and fault identification in welds as the second case study.

The algorithm selected for fault detection in the first two case studies was a combination of *Principal Component Analysis* (PCA), used for data dimensionality reduction, and the *Least Generalized Regression* (LGR) regressor, responsible for determining the presence or absence of faults. The decision to use PCA as a dimensionality reduction tool was based on its ease of implementation and straightforward parallelization, given that the most computationally expensive section of the algorithm consists of a simple matrix multiplication. The choice of LGR was motivated by its low computational cost during inference, enabling rapid execution of the algorithm after determining the regressor's parameters.

In the context of the first two case studies, a new architecture is presented to optimize the real-time fault detection problem in materials and welds through the use of high computational cost monitoring algorithms. Once the parameters representing the fault-free system are loaded into the architecture, the processing of ultrasonic stimuli and responses enables the detection of potential faults in the monitored problem through a regression analysis of the LGR results.

The proposed architecture provides concrete results in accelerating the performance of the PCA+LGR algorithm, achieving a performance gain of at least 12 times compared to the implementation of the algorithm on a commercial ARM processor. The presented architecture utilizes a SoC (*System-on-Chip*) device, which integrates *hardware* (FPGA) and *software* (ARM), for the implementation of the identification model, aiming to maximize computational performance and minimize power consumption. The final validation results

demonstrated high accuracy in significantly less time compared to a conventional microcontroller, even while using a reduced amount of computational resources from the SoC-FPGA chip.

Another method for fault identification can be achieved by comparing the output of an estimated plant with the outputs of the real plant. This strategy is presented in the third case study, which detects potential failures in a pressure sensing system integrated into a real mechanical ventilator. To estimate the mechanical ventilator plant, a *Radial Basis Function* (RBF) neural network was used, which, through a series of experiments, was able to estimate the pressure in the ventilator's main line using airflow and oxygen concentration data.

To assess the effectiveness of the fault identification model in the third case study, a validation of the estimated plant was performed based on the statistical criterion *R-squared* (R²), applied to the model's outputs operating in *Free-Run* (FR) and *One-Step-Ahead* (OSA). In this case, the identified plant presented an R² criterion of 0.66 for FR and 0.82 for OSA. Although the R² values were not high, they were sufficient for the proposed application, as the objective of the study was to perform a simple comparison between the estimated and actual values for fault identification and discontinuities in the pressure sensing system.

Keywords: Fault Detection. System Identification. Mechanical Ventilator. Machine Learning. Embedded Systems.

Listas de figuras

Figura 2.1	Funcionamento simplificado de um PAT (<i>Phased-array transducer</i>) (Benthowave, 2025).	24
Figura 2.2	Experimento do primeiro estudo de caso. Imagem de (Kubrusly <i>et al.</i> , 2016).	25
Figura 2.3	Experimento do segundo estudo de caso. Imagem de (Ferreira <i>et al.</i> , 2025).	26
Figura 2.4	Método de estimativa de falhas utilizado no primeiro e no segundo estudo de caso. Imagem alterada de (Ferreira <i>et al.</i> , 2025).	29
Figura 2.5	Exemplo genérico de detecção de falhas.	30
Figura 2.6	Rede Neural RBF utilizada em (Tonussi; Quintero, 2020).	32
Figura 2.7	Planta simplificada do respirador mecânico TICÊ (Barbalho, 2021)	34
Figura 2.8	Caixa do respirador mecânico TICÊ.	34
Figura 2.9	Pulmão artificial conectado com respirador mecânico TICÊ.	35
Figura 2.10	Esquema de controlador PID profissional apresentado em (BESTARD, 2017)	36
Figura 2.11	Esquema geral de controle de PEEP.	36
Figura 2.12	Loop de controle de O ₂	36
Figura 2.13	Design do preditor Smith.	37
Figura 3.1	Áreas de pesquisa relacionadas com preditores estimativa de estresse + ultrassônico + material.	39
Figura 3.2	Áreas de pesquisa relacionadas com dutos + ultrassônico + material + otimização.	40
Figura 3.3	Áreas de pesquisa relacionadas ao controle de ventiladores mecânicos.	43
Figura 4.1	Esquema de interface do IP-Core HDL proposto. Apresentado em (Mendes <i>et al.</i> , 2023a)	49
Figura 4.2	O <i>Vector Multiplication Module</i> (representado no bloco <i>AXI Stream Wrapper</i> na Fig. 4.1). Apresentado em (Mendes <i>et al.</i> , 2023a)	50
Figura 4.3	Bloco de multiplicação vetorial com <i>pipeline</i> completo.	52
Figura 4.4	Diagrama de alto nível para a arquitetura proposta para monitoramento de dutos.	53
Figura 4.5	Placa SoC utilizada para implementação da arquitetura (Digilent, 2017).	55
Figura 4.6	Respirador mecânico TICE e interface de comunicação com computador.	57
Figura 5.1	Visão de implementação do dispositivo. Esquerda: HDL com 5 <i>cores</i> ; Direita: HDL com 100 <i>cores</i>	60
Figura 5.2	Entradas utilizadas para validação (FR e OSA).	65
Figura 5.3	Estimação da pressão PT2 OSA.	66
Figura 5.4	Correlação móvel OSA (janela de 200 dados).	66
Figura 5.5	Estimação da pressão PT2 FR.	67

Figura 5.6 Correlação móvel FR (janela de 200 dados).	67
Figura 5.7 Modelo no simulink com hiper-parâmetros NARX definidos (na=3, nb=2).	
68	
Figura 5.8 Esquemático do design da rede neural RBF elaborada por (Sampaio <i>et al.</i>, 2016).	68

Lista de tabelas

Tabela 3.1 Artigos sobre estimação de estresse, selecionados por relevância (número de citações)	39
Tabela 3.2 Artigos sobre monitoramento de dutos, selecionados por relevância (número de citações)	41
Tabela 3.3 Resultados da pesquisa bibliográfica para a arquitetura implementada . .	42
Tabela 3.4 Trabalhos principais relacionados com respiração mecânica com alto índice de citação.	44
Tabela 3.5 Resultados da pesquisa bibliográfica para o respirador mecânico. (1/2) . .	46
Tabela 3.6 Resultados da pesquisa bibliográfica para o respirador mecânico. (2/2) . .	47
Tabela 4.1 Tabela de recursos da placa ZYBO Z-20.	55
Tabela 5.1 Utilização de <i>hardware</i> da arquitetura geral e dos módulos RTL implementados no chip xc7z020clg400-1.	60
Tabela 5.2 Latência computacional medida	61
Tabela 5.3 Resumo do relatório de potência.	61
Tabela 5.4 Resumo do relatório de consumo de energia	62
Tabela 5.5 Utilização de <i>hardware</i> do design implementado	62
Tabela 5.6 Resultados experimentais da implementação de <i>hardware</i>	63

Lista de abreviaturas e siglas

LGR	Least Generalized Regression
MIMO	<i>Multiple Inputs Multiple Outputs</i>
MISO	<i>Multiple Input Single Output</i>
P-PID	<i>Professional Proportional Intergal Derivative Controller</i>
PCV	<i>Pressure Controlled Ventilation</i>
PEEP	<i>Positive end-expiratory pressure</i>
R2	<i>R-squared correlation</i>
RBF	<i>Radial Basis Function</i>
SISO	<i>Single Input Single Output</i>
UnB	Universidade de Brasília
UPE	Universidade de Pernambuco
VCV	<i>Volume Controlled Ventilation</i>

Sumário

1	Introdução	16
1.1	Contexto da Proposta	16
1.1.1	Contexto Sobre o Primeiro Estudo de Caso	16
1.1.2	Contexto Sobre o Segundo Estudo de Caso	17
1.1.3	Contexto Sobre o Terceiro Estudo de Caso	17
1.2	Hipóteses da Proposta	18
1.3	Objetivos	19
1.3.1	Objetivos Gerais	19
1.3.2	Objetivos Específicos	19
1.4	Aspectos Metodológicos do Trabalho	20
1.5	Contribuições da proposta	21
1.6	Estrutura do Texto	21
2	Fundamentação Teórica	23
2.1	Técnicas de Detecção de Falhas em Materiais Sobre Estresse	23
2.1.1	Uso de Ultrassom em Monitoramento	24
2.1.2	Experimento do Primeiro Estudo de Caso	24
2.2	Técnicas de Detecção de Falhas em Dutos e Tubulações	25
2.2.1	Experimento do Segundo Estudo de Caso	25
2.3	Algoritmos de Redução de Dimensionalidade	26
2.3.1	Algoritmo ICA	26
2.3.2	Algoritmo PCA	27
2.4	Algoritmo LGR	28
2.5	Detecção de Falhas em Sistemas Desconhecidos	30
2.6	NARX em Modelagem de Sistemas	31
2.7	Rede Neural RBF	31
2.8	Métodos de validação utilizados: <i>Free-Run</i> e <i>One-Step-Ahead</i>	32
2.9	Terceiro Estudo de Caso - Respirador TICÊ	34
2.10	Controle do Respirador TICÊ	35
2.11	Considerações finais do capítulo	37
3	Revisão da Literatura	38
3.1	Revisão da Literatura para o Primeiro e o Segundo Estudo de Caso	38
3.1.1	Revisão Geral da bibliografia de arquiteturas para detecção de falhas	41
3.2	Revisão da Literatura para o Terceiro Estudo de Caso	42
3.2.1	Sobre Controle e Correção de Falhas em Ventiladores Mecânicos . .	42
3.2.2	Revisão Geral da bibliografia do terceiro estudo de caso	44

3.3	Considerações finais do capítulo	48
4	Metodologia	49
4.1	Arquitetura Desenvolvida para o Primeiro Estudo de Caso	49
4.2	Melhorias da Arquitetura Para Sua Utilização no Segundo Estudo de Caso	51
4.2.1	Implementação puramente em SW (ARM) e <i>code-profiling</i>	53
4.2.2	Plataformas Computacionais usadas para Avaliação de Desempenho	54
4.3	Terceiro Estudo de Caso - Respirador TICÊ	55
4.3.1	Supervisório de Alto Nível para Estímulo e Monitoramento do Respirador –SANER	56
4.3.2	Hiper-parâmetros NARX do modelo de pressão sintética	57
4.3.3	Ensaio para identificação do sistema de pressão sintética	58
4.4	Considerações finais do capítulo	58
5	Resultados	60
5.1	Primeiro e Segundo Estudo de Caso - Arquitetura HW/SW Co-design	60
5.1.1	Desempenho da Arquitetura Proposta para o Primeiro Estudo de Caso	60
5.1.2	Desempenho da Arquitetura Proposta para o Segundo Estudo de Caso	62
5.2	Terceiro Estudo de Caso - Respirador Mecânico	63
5.2.1	Dados disponíveis do sistema respirador mecânico	63
5.2.2	Identificação do sistema de pressão sintética	64
5.2.3	Proposta de Implementação Embarcada do Terceiro Estudo de Caso	68
5.3	Discussões sobre os Resultados	69
5.3.1	Resultados do Primeiro Estudo de Caso	69
5.3.2	Resultados do Segundo Estudo de Caso	69
5.3.3	Resultados do Terceiro Estudo de Caso	70
6	Conclusões	71
6.1	Primeiro e Segundo Estudo de Caso	71
6.2	Terceiro Estudo de Caso	72
6.3	Trabalhos Futuros	72
Referências	74	
Anexo A Trabalhos Publicados	85	

1 Introdução

Estratégias preditivas de manutenção e detecção de falhas geram uma economia de 30% a 40% nos gastos anuais das empresas de energia americanas, de acordo com dados do Departamento de Energia dos Estados Unidos (Program, 2010). Além disso, a detecção antecipada de falhas em sistemas críticos contribui significativamente para a redução de situações de alto risco à vida humana. Neste trabalho, são investigadas otimizações em estratégias de detecção de falhas, aplicadas a sistemas críticos reais. Por meio de implementações em hardware, essas estratégias de alto custo computacional tornam-se viáveis para execução em tempo real e em plataformas embarcadas.

Esse trabalho utiliza o formato de estudos de caso pra apresentar diferentes soluções em hardware de otimização de algoritmos já conhecidos na indústria, aplicados para detecção de falhas. Três estudos de caso são apresentados: Monitoramento de estruturas tensionadas, monitoramento da integridade estrutural de oleodutos e o monitoramento de anomalias em um sensor médico de pressão inspiratória.

1.1 Contexto da Proposta

1.1.1 Contexto Sobre o Primeiro Estudo de Caso

O monitoramento em tempo real de estruturas é um tópico de interesse atual na engenharia, devido ao seu potencial de redução de riscos de falhas catastróficas (Yang *et al.*, 2023). Dentre as técnicas de monitoramento, métodos não destrutivos vêm ganhando notoriedade, em especial o monitoramento por ondas ultrassônicas (Ji *et al.*, 2021). Essas ondas são utilizadas para medir estresse mecânico através das variações de resposta do material quando submetido a um estímulo de onda de ultrassom, pelo efeito acústico-elástico (Pao; Sachse; Fukuoka, 1984).

O monitoramento proativo de estruturas promove uma mudança na abordagem clássica de manutenções realizadas em intervalos de tempo fixos, reduzindo as margens de erro humano nas inspeções e permitindo que as falhas estruturais sejam tratadas de forma preventiva, em vez de reativa (Jia; Li, 2023).

Entre as soluções de monitoramento em tempo real, existem opções parcialmente embarcadas que possibilitam a leitura de sensores integrados e a transmissão dos dados, mas não são projetadas para processar os resultados localmente. Em vez disso, essas soluções utilizam uma análise *offline* em um sistema integrado separado do sistema embarcado, devido ao alto custo computacional dos algoritmos empregados (Barile *et al.*, 2018).

Alguns dos algoritmos modernos mais utilizados para a análise de dados estruturais empregam redes neurais profundas (*deep learning*), que normalmente são inviáveis para

implementação em sistemas computacionais compactos (Lim; Sohn, 2020), (Vieira; Lambros, 2021) e (Zhang *et al.*, 2022). No trabalho de (Ferreira *et al.*, 2025), uma versão alternativa da computação desses algoritmos é proposta, através da utilização de uma combinação do redutor de dimensionalidade *Principal Component Analysis* (PCA) com o regressor *Least Generalized Regression* (LGR). Neste trabalho, explora-se a otimização do monitoramento por PCA+LGR, considerando que o PCA, devido à sua implementação simples e paralelizável na etapa de maior custo computacional, é um forte candidato à aceleração.

1.1.2 Contexto Sobre o Segundo Estudo de Caso

O monitoramento da integridade estrutural de oleodutos também é explorado neste trabalho, devido à gravidade dos riscos associados às suas falhas (Ferreira, 2022). Neste trabalho, uma solução otimizada com foco no monitoramento de falhas em soldas desses oleodutos é explorada, utilizando abordagens similares a aquelas apresentadas no primeiro estudo de caso. Para o segundo estudo de caso, o monitoramento em pontos de solda foi escolhido tendo em vista que é a região da tubulação com maior vulnerabilidade estrutural (Shaloo *et al.*, 2022).

Para o monitoramento desses dutos, é comum encontrar estratégias de detecção de falhas através do processamento de dados de sensores ultrassônicos (*B-Scan*), permitindo classificar e detectar falhas estruturais de maneira automática (Fan; Bai; Chen, 2024). A utilização desses métodos em conjunto com *deep learning* é explorada na literatura, através da utilização e algoritmos similares aos apresentados no primeiro estudo de caso. Da mesma forma, soluções como a utilização do PCA+LGR como alternativa para a implementação em *deep-learning* também são aplicáveis ao problema.

Assim como no primeiro estudo de caso, as técnicas de monitoramento de oleodutos estão em constante evolução. Dentre elas, as técnicas de monitoramento em tempo real têm ganhado destaque, pois permitem uma redução significativa nos custos de manutenção. Isso ocorre devido à diminuição do tempo de exposição a potenciais falhas estruturais, evitando que essas falhas evoluam para problemas mais graves ou catastróficos (Ferreira *et al.*, 2022).

Para viabilizar a utilização dos algoritmos de monitoramento de tubulações em tempo real, soluções similares às apresentadas no primeiro estudo de caso foram aplicadas.

1.1.3 Contexto Sobre o Terceiro Estudo de Caso

O custo de um respirador mecânico importado pode ser estimado na ordem de quase 150 mil reais. Nesse contexto, o projeto FINATEC TICÊ propõe a construção de um respirador nacional de baixo custo, desenvolvido com o objetivo de alcançar um custo unitário na faixa de 30 a 60 mil reais, um preço consideravelmente menor que outras opções disponíveis no mercado. Por conta disso, alguns componentes do respirador foram escolhidos de maneira mais econômica, prezando pelo custo-benefício. Dito isso, uma das maiores desvantagens

da utilização de componentes de baixo custo, é o risco de produção de dados errôneos no sistema de sensoriamento do respirador. Para alcançar um baixo custo de fabricação no desenvolvimento do respirador TICÊ, não foram incorporadas estratégias complexas de detecção de perda de integridade de sensores. Em outras palavras, não foram utilizados sensores redundantes nas linhas de controle, nem implementados algoritmos de monitoramento de sanidade de sinais em tempo real.

Garantir a integridade desses dados utilizados para o controle do respirador mecânico é fundamental para assegurar a segurança do paciente. Apesar dos componentes individuais utilizados oferecerem um nível razoável de integridade, pode ser necessário雇用 estratégias de identificação de falhas, como a detecção de dados errôneos por monitoramento *online* dos sensores.

Na literatura de tratamento de sinais biomédicos de alta integridade (sinais que possuem baixas taxas de dados errôneos), é comum employar estratégias de verificação de sanidade através da comparação de sinais sintéticos (estimados por modelos) e sinais reais (Azizi, 2024), (Foraker *et al.*, 2020). Essas estratégias permitem a invalidação de sinais errôneos, evitando que eles sejam utilizados incorretamente em malhas de controle críticas para o funcionamento de equipamentos biomédicos (Azizi, 2024).

Durante o projeto de desenvolvimento do respirador mecânico TICÊ, para permitir a detecção de possíveis falhas do sensor de pressão real, foi feita a comparação entre dados estimados de pressão e os dados mensurados pelo sensor. Para isso, foi realizada uma identificação não determinística da planta na linha inspiratória do respirador mecânico. Comparando com outros trabalhos, em (Sharifi; Chbat, 2015) e (Tran *et al.*, 2021) foram apresentadas identificações polinomiais determinísticas do modelo de um respirador, o que difere da modelagem proposta neste trabalho, que utiliza uma abordagem não determinística baseada em uma rede neural para a identificação do sistema não linear do respirador. Apesar do trabalho (Guo *et al.*, 2023) apresentar uma solução moderna e não determinística para a identificação do sistema de um respirador mecânico, a identificação foi realizada em um modelo conhecido e puramente teórico. Em contrapartida, este trabalho apresenta uma solução na qual o modelo proposto foi testado e validado em uma bancada com um respirador mecânico real acoplado a um pulmão artificial.

1.2 Hipóteses da Proposta

O modelo escolhido para a identificação de falhas do primeiro e do segundo estudo de caso deste trabalho foi o modelo de monitoramento de estruturas apresentado em (Virkkunen *et al.*, 2021), otimizado através do uso do PCA, conforme apresentado em (Holguin; Ayala; Kubrusly, 2021). A **primeira hipótese** proposta é que é possível realizar a paralelização das partes do algoritmo com os maiores custos computacionais, justificando, de maneira objetiva, o desenvolvimento de uma nova arquitetura através da estratégia de HW/SW co-

design, focada na aceleração das etapas paralelizáveis, de forma análoga ao que foi feito nos trabalhos de (Mendes *et al.*, 2023b) e (Xu *et al.*, 2022).

Uma vez definida uma nova arquitetura para a execução do algoritmo de identificação de falhas, uma **segunda hipótese** considerada foi que uma arquitetura dedicada em HW/SW *co-design* apresentaria um baixo custo energético para calcular a inferência de dados quando comparada a um processador convencional (Irmak; Alachiotis; Ziener, 2021).

No contexto de uma arquitetura dedicada, a **terceira hipótese** é que uma arquitetura dedicada viabilizaria a execução em tempo real de algoritmos de monitoramento de estruturas.

Para o caso do respirador (terceiro estudo de caso), nenhuma estratégia de validação de sinais foi empregada. Com o objetivo de melhorar a integridade dos sinais do respirador mecânico, uma **quarta hipótese** foi proposta: que é possível estimar um sinal sintético de pressão na linha inspiratória do paciente, permitindo em teoria, a aplicação de uma estratégia de validação de sinais por comparação entre os sinais de pressão real e os sinais de pressão sintética, aumentando significativamente a integridade dos dados linha principal de controle.

1.3 Objetivos

1.3.1 Objetivos Gerais

O objetivo geral é desenvolver soluções embarcáveis para técnicas de identificação de falhas em três estudos de casos diferentes: Detecção de falhas em materiais tensionados, detecção de falhas em soldas de tubulações, detecção de falhas em sensores de pressão aplicados para respiração mecânica.

1.3.2 Objetivos Específicos

1. Desenvolver uma solução embarcada do algoritmo de monitoramento apresentado em (Ferreira *et al.*, 2025), usando um processador ARM *standalone* com arquitetura *single-core* e *dual-core*.
2. Desenvolver uma arquitetura em hardware eficiente para aceleração do algoritmo PCA+LGR, utilizando estratégias de controle sequencial de dados baseado em máquinas de estados finitos.
3. Melhorar o desempenho da implementação em hardware do algoritmo PCA+LGR, utilizando arquiteturas do tipo *pipeline*.
4. Desenvolver um modelo de identificação de falhas para sistema de sensoriamento de pressão de um respirador mecânico, baseado na comparação entre o sistema identificado e os dados de pressão obtidos pelo sistema de medição.

1.4 Aspectos Metodológicos do Trabalho

1. A plataforma selecionada para a implementação da arquitetura foi um SoC composto por um ARM e uma FPGA, interconectados via barramento AXI4. A escolha de um SoC baseado em FPGA, em vez de um SoC com GPU como plataforma de aceleração de hardware, deve-se ao seu menor consumo energético. Um SoC FPGA da classe ZYNQ, por exemplo, opera com um consumo de aproximadamente 1-2 W, enquanto uma placa SoC com GPU de classe similar, como a Jetson Nano, apresenta um consumo estimado entre 5-10 W. Dessa forma, a adoção do SoC FPGA justifica-se pelo seu baixo consumo energético, possibilitando embarcar a arquitetura proposta.
2. A arquitetura proposta para o primeiro estudo de caso (identificação de falhas de estruturas sob estresse mecânico) está baseada em máquina de estados finitos, prezando pela simplicidade de implementação. A implementação da arquitetura em máquina de estados permite uma maior facilidade de *debug* durante o desenvolvimento inicial, assim como uma maior facilidade para atender os requisitos de *timing* da arquitetura.
3. No segundo estudo de caso (identificação de falhas em soldas de tubulações), a arquitetura proposta passou por aprimoramentos no design, considerando as limitações do uso de uma máquina de estados finitos. A nova versão adota uma estrutura pipeline para aumentar o *throughput* do algoritmo acelerado em FPGA. Essa abordagem não apenas reduz o tempo de processamento, mas também facilita a integração entre o PS e o PL, permitindo o envio contínuo de dados após a latência inicial.
4. Para o terceiro estudo de caso (identificação de falhas no sensor de pressão de um respirador mecânico), uma nova interface de comunicação com o respirador mecânico foi necessária de ser desenvolvida. Por conta das limitações da interface padrão do respirador, não era possível executar ensaios com estímulos de entrada controlada por *scripts*. Por conta disso, uma nova interface de comunicação e monitoramento dos dados do respirador foi desenvolvida em MATLAB, comunicando com o respirador através de uma interface USB.
5. Uma vez desenvolvida a ferramenta de comunicação com o respirador, foi identificado o modelo do sistema de sensoriamento de pressão inspiratória. Esse modelo foi utilizado para viabilizar a identificação de possíveis falhas no sensor de pressão. Para explorar a implementação embarcada do terceiro caso se estudo, foi apresentada uma arquitetura já desenvolvida em outro trabalho ([Tonussi; Quintero, 2020](#)).
6. Para otimizar os hiper-parâmetros do modelo apresentado no terceiro estudo de caso, foi desenvolvido um algoritmo de teste recursivo. Esse algoritmo busca, por força bruta, as combinações que resultam nos melhores valores de correlação entre a saída real e a saída estimada.

1.5 Contribuições da proposta

O desenvolvimento apresentado no primeiro e segundo estudo de caso resultou em uma nova arquitetura de *hardware* para detecção de falhas, aplicada para os problemas apresentados em (Virkkunen *et al.*, 2021) e (Kubrusly *et al.*, 2016). Através dessa arquitetura, um sistema embarcado monitoramento de estruturas em tempo real pode ser desenvolvido, capaz de detectar falhas e anomalias por estresse mecânico. Dessa forma, tarefas que comumente são executadas por manutenção programada, podem ser substituídas por um dispositivo embarcado, reduzindo a necessidade de tarefas manuais sem comprometer a segurança do sistema.

Como resultado, este trabalho contribuiu para a publicação de dois artigos: o primeiro sobre a comparação de desempenho entre HDL e HLS na identificação de estresse em materiais (Mendes *et al.*, 2023a), e o segundo sobre a detecção de falhas em tubulações de oleodutos por meio de *machine learning*, atualmente em revisão pela *IEEE Transactions on Industrial Informatics* (Ferreira *et al.*, 2025). Detalhes dessas publicações podem ser encontrados no Anexo A.

Neste trabalho, também são apresentados alguns resultados que foram coletados ao longo do desenvolvimento do respirador de baixo custo TICÊ, projeto que recebeu atenção dos canais de mídia nacionais (G1 Globo, 2022). Durante o desenvolvimento do mesmo, uma aplicação prática de identificação de falha foi explorada, especificamente, uma modelagem não determinística do modelo que relaciona pressão e fluxo permitiu a detecção de possíveis falhas na medição da linha inspiratória do respirador.

1.6 Estrutura do Texto

Este documento está estruturado em seis capítulos:

No Capítulo 2, é apresentada a fundamentação teórica, introduzindo os conceitos fundamentais para o entendimento do trabalho, entre eles: Detecção e falhas em sistemas desconhecidos, o algoritmo PCA, a modelagem de sistemas por parâmetros NARX, o conceito de uma rede neural RBF, o respirador mecânico utilizado e os métodos de validação OSA e *free-run*.

No Capítulo 3, é apresentado um estudo bibliográfico sobre o estado da arte na área de monitoramento estrutural por estratégias computacionais, destacando os dois primeiros estudos de caso escolhidos para aceleração em *hardware*. Também é apresentado um estudo sobre o estado da arte na área de identificação de sistemas em respiradores mecânicos.

No Capítulo 4, é apresentada a metodologia utilizada para alcançar os objetivos propostos, evidenciando os algoritmos e as arquiteturas em *hardware/software* desenvolvidas.

No Capítulo 5, são apresentados os resultados, validando os três estudos de caso e apresentando a arquitetura desenvolvida em *hardware* para os dois primeiros estudos de caso

em conjunto com uma proposta em hardware para o terceiro estudo de caso.

No Capítulo 6, são discutidos a conclusão e a confirmação das hipóteses a partir dos resultados apresentados no Capítulo 5.

2 Fundamentação Teórica

2.1 Técnicas de Detecção de Falhas em Materiais Sobre Es-tresse

Através do monitoramento de falhas em materiais sob tensão, é possível prevenir acidentes graves decorrentes de rupturas e falhas críticas, aumentando a vida útil dos materiais e reduzindo tanto os riscos financeiros quanto os impactos à segurança humana. Esse monitoramento pode ser classificado em duas categorias: destrutivo e não destrutivo.

- Monitoramento Destrutivo
 - Teste de Fadiga: Ensaio destrutivo de fadiga de materiais, através de teste repetitivo de ductibilidade ([ASTM International, 2015](#)).
 - Teste de Tensão: Ensaio destrutivo de tensão, através de aplicação de uma carga de tensão até o limite da deformação elástica do material ([ASTM International, 2016](#)).
 - Teste Fratura Mecânica: Estudo de comportamento de trincas em materiais sobre teste ([ASTM International, 2020](#)).
- Monitoramento Não-Destrutivo
 - Inspeção Visual: Método de inspeção manual simples, alto custo e suscetível a erro humano.
 - Teste de Radiografia: Utiliza raios-X para detecção de possíveis trincas e falhas mecânicas no material, normalmente acompanhada de uma inspeção manual visual ([Hellier, 2012](#)).
 - Teste Ultrassônico: Utiliza de ondas de alta frequência para identificação de falhas internas e superficiais de materiais ([Krautkrämer; Krautkrämer, 1990](#)).

Apesar dos ensaios destrutivos proporcionarem uma boa estimativa do tempo de falha de alguns materiais, fatores adversos como clima, ambiente e variações no processo de fabricação podem impedir que o material utilizado como corpo de prova seja verdadeiramente representativo do material a ser monitorado. Por conta disso, métodos não destrutivos podem ser preferíveis para aplicações críticas, em especial os métodos por monitoramento por raio-X e por ondas ultrassônicas. Para o primeiro estudo de caso, uma abordagem não destrutiva foi escolhida, mais especificamente, o monitoramento por ultrassom, escolhido por possuir baixo custo de operação quando comparado com monitoramento por Raio-X. A seguinte subseção apresenta mais detalhes sobre o monitoramento por ultrassom e o estudo de caso abordado nesse trabalho.

2.1.1 Uso de Ultrassom em Monitoramento

Ondas ultrassônicas são extensivamente utilizadas na área de monitoramento de materiais de maneira não destrutiva. Dentre as técnicas mais utilizadas, estão o monitoramento por ondas *Lamb* e o monitoramento por ondas *B-Scan*.

- **Monitoramento por ondas *Lamb*:** O monitoramento por ondas *Lamb* funciona através da excitação de um material rígido com ondas ultrassônicas (usando um transdutor piezoelétrico, por exemplo). As oscilações do material e seus modos de vibração são então capturados por um sensor adequado, como um *strain gauge*. Uma característica fundamental da técnica de monitoramento por ondas *Lamb* é que essas ondas são restritas a apenas um "plano" do material, apresentando resultados mais satisfatórios em materiais de baixa espessura (Rose, 2014).
- **Monitoramento por ondas *B-Scan*:** O monitoramento por ondas *B-Scan* ocorre por meio do processamento das reflexões das ondas ultrassônicas para a formação de imagens. Para gerar essas ondas, é necessária a utilização de um transdutor, que pode ser de um único elemento de transmissão/recepção ou um transdutor composto por uma matriz controlável de elementos (*Phased-array transducer*) (Figura 2.1).

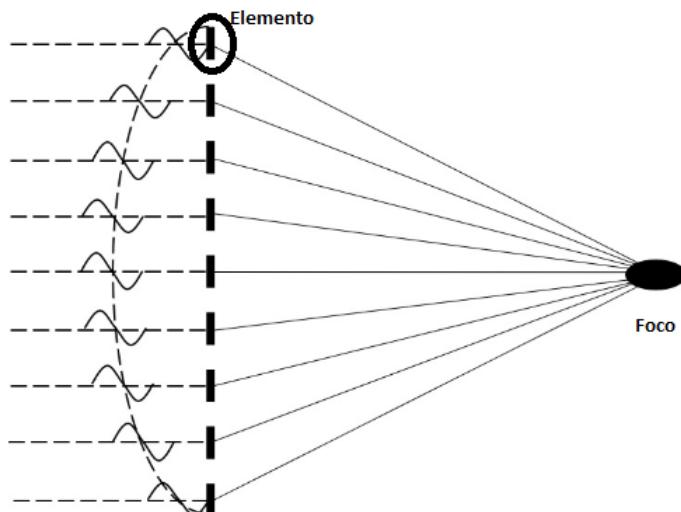


Figura 2.1 – Funcionamento simplificado de um PAT (*Phased-array transducer*) (Benthowave, 2025).

Para o primeiro estudo de caso, ondas do tipo *Lamb* foram utilizadas para a coleta de dados, já no segundo estudo de caso, leituras por ondas *B-Scan* foram utilizadas.

2.1.2 Experimento do Primeiro Estudo de Caso

Os dados utilizados para o estudo de estimativa de estresse em materiais foram coletados do estudo (Kubrusly *et al.*, 2016), nesse estudo, uma placa de alumínio com dimensões de $800 \times 100 \times 3$ mm possui as suas extremidades mais distantes fixadas em uma

estrutura onde uma extremidade estava fixa enquanto a outra se movia sob a ação de um parafuso sem-fim. Para criar um sinal de perturbação na placa, um transdutor piezoelétrico foi utilizado para gerar um pulso de banda larga com largura de banda de 2.5 MHz, induzindo a propagação de múltiplos modos de *Lamb* na placa, os quais foram registrados por um transdutor piezoelétrico situado a 700 mm de distância do transmissor. Cargas de tensão foram aplicadas na placa ao girar o parafuso sem-fim, promovendo deformações entre 0 a 150 $\mu\text{m}/\text{m}$. Essas deformações foram registradas por um *strain-gauge* resistivo localizado no centro da placa (Figura 2.3). Por fim, o processo de excitação da placa foi realizado 499 vezes para diferentes níveis de deformação dentro da faixa mencionada. Os sinais adquiridos foram amostrados a 10 MHz, com 8.192 amostras, abrangendo o intervalo observável de 130,0 μs a 949,1 μs (Mendes *et al.*, 2023a).

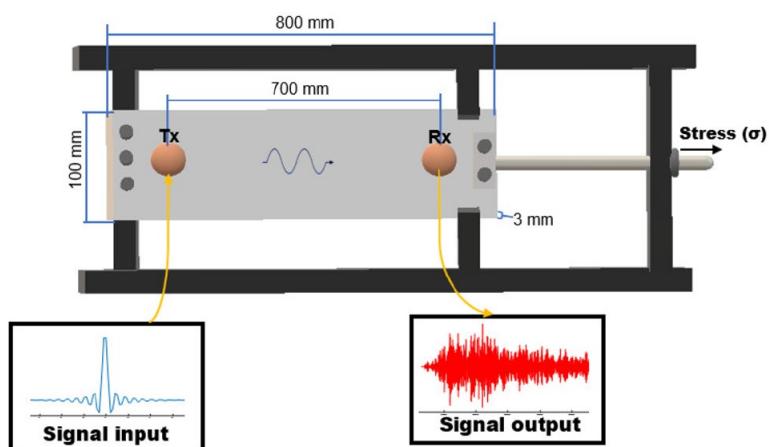


Figura 2.2 – Experimento do primeiro estudo de caso. Imagem de (Kubrusly *et al.*, 2016).

2.2 Técnicas de Detecção de Falhas em Dutos e Tubulações

Para o monitoramento de falhas em tubulações, as técnicas destrutivas apresentadas em 2.1 também podem ser aplicadas, embora com menor efetividade. Isso ocorre porque as tubulações possuem características estruturais distintas dos materiais sólidos utilizados nos ensaios, com diferentes pontos suscetíveis a falhas. Um exemplo comum de falha em tubulações são os pontos de solda, que foram escolhidos como foco do segundo estudo de caso. Para o monitoramento dessas falhas, é comum a utilização das mesmas técnicas não destrutivas apresentadas em 2.1.

2.2.1 Experimento do Segundo Estudo de Caso

Os dados desse estudo de caso foram coletados através de um experimento: O estudo analisou uma solda em forma de U com 30 mm de largura, realizada em um tubo de aço inoxidável austenítico 316L. O tubo possuía 40 mm de espessura e aproximadamente 400

mm de diâmetro externo. Nesse tubo, foram identificadas três trincas planas circunferenciais próximas à raiz da solda com profundidades de 1,6 mm, 4,0 mm e 8,6 mm, geradas por fadiga térmica (Figura 2.2).

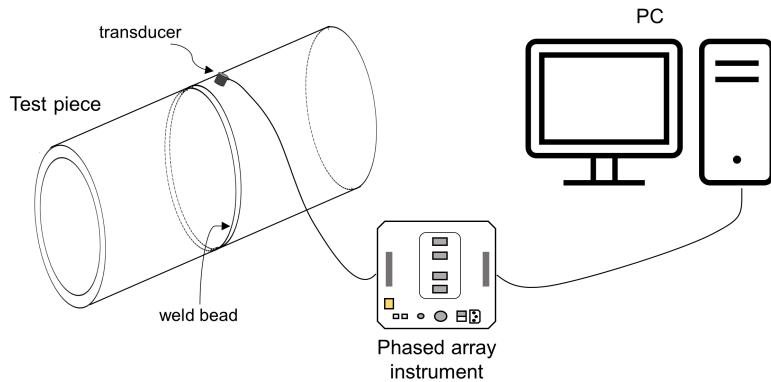


Figura 2.3 – Experimento do segundo estudo de caso. Imagem de (Ferreira *et al.*, 2025).

Através de leituras ultrassônicas foram coletados dados utilizando uma varredura *phased array* com ondas cisalhantes (Transmission Receive Shear), utilizando sondas matriciais de 1,8 MHz. Para otimizar a visualização das trincas, foi aplicado um único ângulo de varredura de 45°. As informações ultrassônicas foram registradas no formato B-scan, mais detalhes sobre o tubo e os métodos estão disponíveis em (Virkkunen *et al.*, 2021).

Devido à alta dimensionalidade dos dados representados pelas imagens ultrassônicas (UT-images), foi utilizado um procedimento de re-amostragem como etapa de pré-processamento. As imagens B-scan, originalmente com resolução de 256 x 256 pixels, foram redimensionadas para 128 x 128 pixels. Essa resolução foi selecionada para garantir que as informações visuais fossem preservadas após o redimensionamento.

2.3 Algoritmos de Redução de Dimensionalidade

2.3.1 Algoritmo ICA

A Análise de Componentes Independentes (ICA) é um algoritmo que permite a separação de sinais de fontes diferentes, através da análise da dependência estatística entre os dados. Além da capacidade de separação de dados, o ICA também pode ser aplicado para aplicações de redução de dimensionalidade, como no trabalho de (Lopes, 2013), em que ICA é utilizado para extração de características principais em sinais de ECG.

O algoritmo do ICA pode ser simplificado da seguinte maneira:

$$\mathbf{X} = \mathbf{A} \cdot \mathbf{S}$$

- \mathbf{X} é a matriz $m \times n$ dos sinais observados, onde m é o número de sinais e n o número de amostras.
- \mathbf{S} é a matriz $k \times n$ das fontes independentes, onde k é o número de fontes.
- A é a matriz $m \times k$ de mistura, que é desconhecida.

Como saída do ICA, obtém-se a matriz \mathbf{S} a partir das observações \mathbf{X} , ou seja, basta encontrar a matriz W :

$$\mathbf{S} = W \cdot \mathbf{X}$$

W é a matriz de separação, que é a inversa de A .

Em resumo, O ICA tenta reverter a mistura linear $X = A \cdot S$, buscando encontrar a matriz W que recupera as fontes S independentes.

Apesar da capacidade do ICA de redução de dimensionalidade de dados, a principal desvantagem desta técnica é que não ordena os componentes em termos da variância dos dados. Nesse sentido, o algoritmo PCA é mais apropriado, pois, facilita a seleção dos componentes mais representativos dos conjuntos de dados do primeiro e segundo estudos de caso. Este algoritmo é explicado em detalhes na seguinte subseção.

2.3.2 Algoritmo PCA

Tendo como objetivo utilizar um método eficiente de detecção de falhas, o algoritmo Análise de Componentes Principais (PCA) foi escolhido como ferramenta principal de redução de dimensionalidade dos problemas explorados. O algoritmo PCA foi escolhido por conta de sua simplicidade de implementação e por sua capacidade de paralelização e aceleração em hardware, explorado em mais detalhes na Seção 4.2.1. Em resumo, o PCA foi utilizado para reduzir a dimensionalidade de dados de sistemas não lineares desconhecidos, permitindo uma redução significativa do tamanho necessário do conjunto de dados a ser utilizado para os modelos de identificação de falhas apresentados neste trabalho, sem perdas significativas de informações.

O PCA funciona através do cálculo de uma transformação linear truncada (Eq. 2.1), em que K colunas da matriz \mathbf{V} (ou seja, os componentes principais, PCs) formam uma base ortogonal para as K características, proporcionando uma aproximação de baixa dimensionalidade (\mathbf{T}_K) dos dados de alta dimensionalidade \mathbf{X} . Essa transformação está associada a outra fatoração matricial dada pela Decomposição em Valores Singulares (SVD) de $\mathbf{X} \in \mathbb{C}^{n \times m}$ (Eq. 2.2).

$$\mathbf{T}_K = \mathbf{X}\mathbf{V}_K \quad (2.1)$$

$$\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^H, \quad (2.2)$$

onde $\mathbf{U} \in \mathbb{C}^{n \times n}$ e $\mathbf{V} \in \mathbb{C}^{m \times m}$ são matrizes unitárias, e \mathbf{V} possuem uma base ortonormal dos autovalores correspondentes (Brunton; Kutz, 2019). H denota o transposto conjugado complexo.

A PCA possui duas etapas principais: a etapa de treinamento e a etapa de projeção. Durante a etapa de treinamento, os componentes principais são obtidos utilizando a SVD. Essa etapa é computacionalmente complexa e demorada, pois envolve o cálculo de uma fatoração matricial (Bengio; Courville; Vincent, 2013). A etapa de projeção exige os componentes principais (PCs) e os valores de entrada para realizar a redução de dimensionalidade (Eq. 2.1).

Neste trabalho, o PCA foi utilizado em dois estudos de caso diferentes:

- No primeiro estudo de caso, que utilizou dados do estudo (Kubrusly *et al.*, 2016), descrito em detalhes na Seção 2.1.2.
- No segundo estudo de caso, que utilizou os dados do estudo (Virkkunen *et al.*, 2021), descrito em detalhes na Seção 2.2.1.

Uma vez obtida a matriz reduzida pelo PCA (T_K), é possível de executar a detecção da existência de falhas do modelo através da utilização de qualquer algoritmo de classificação. Neste trabalho, foi explorada a utilização do algoritmo LGR, por conta de sua simplicidade de implementação e sua boa performance de estimativa (Ferreira *et al.*, 2025). A Seção seguinte apresenta com mais detalhes o funcionamento do algoritmo, utilizando os segundo estudo de caso como exemplo.

2.4 Algoritmo LGR

No primeiro e no segundo estudo de caso, foi utilizado o algoritmo LGR (*Least Generalized Regression*) como classificador final para a detecção de falhas, depois da execução do redutor de dimensionalidade PCA (Seção anterior).

No primeiro estudo de caso, o LGR foi implementado para a identificação binária de falhas plásticas em materiais metálicos, já no segundo estudo de caso, o algoritmo foi implementado para a identificação binária de falhas em soldas.

O algoritmo LGR implementado (Algoritmo 2.1), consiste de uma simples regressão e classificação dos valores da matrix reduzida computada pelo PCA, utilizando valores de *coef* e *bias* pré determinados:

Algoritmo 2.1 Regressão LGR (exemplo segundo estudo de caso)

- 1: **Entrada:** Matriz de coeficientes LGR (coef), Linha de dados reduzida pelo PCA (Xred_row), bias LGR (bias)
- 2: **Saída:** Classificação em dano da tubulação
- 3: Aplicação dos coeficientes LGR:

$$arg = coef \cdot Xred_row$$

- 4: Adição de bias:

$$arg = arg + bias$$

- 5: Cálculo de probabilidade \hat{p} utilizando a função sigmoide:

$$\hat{p} = \frac{1}{1 + e^{-arg}}$$

- 6: Classificação com base no limite pré definido:

$$Y = \begin{cases} 1 & \text{se } \hat{p} < 0.5 \quad (\text{duto danificado}) \\ 0 & \text{se } \hat{p} \geq 0.5 \quad (\text{duto não danificado}) \end{cases}$$

A utilização do PCA como método de redução de dimensionalidade e o LGR como regressor permite a estimativa das falhas de maneira direta, de acordo com o diagrama apresentado na Figura 2.4.

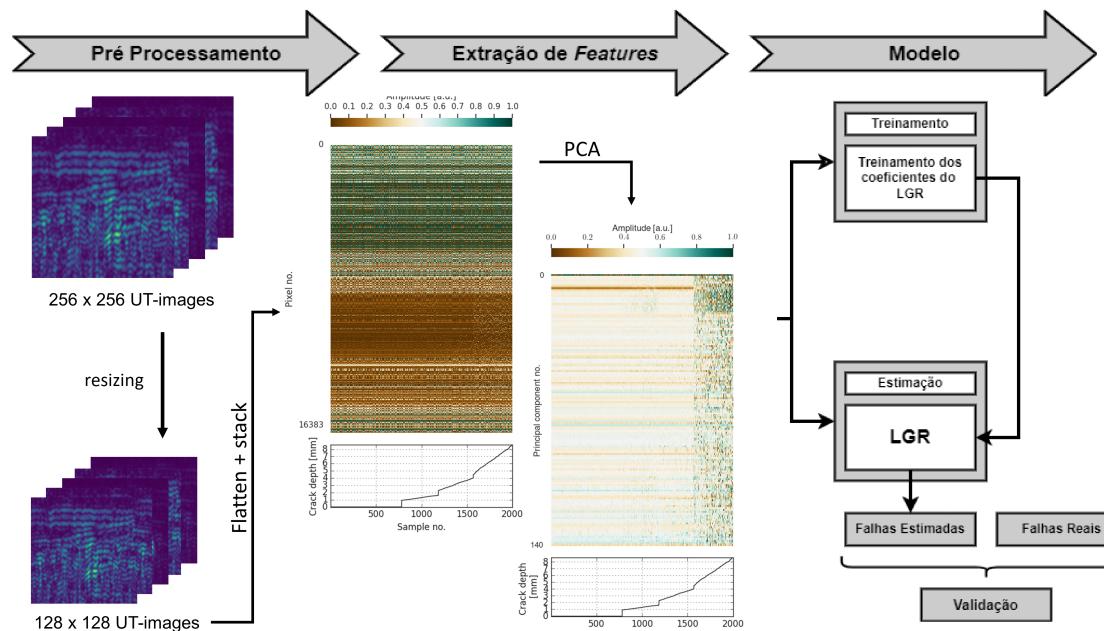


Figura 2.4 – Método de estimativa de falhas utilizado no primeiro e no segundo estudo de caso.
Imagen alterada de ([Ferreira et al., 2025](#)).

É importante ressaltar que, neste trabalho, o objetivo do primeiro e segundo estudo de caso é viabilizar a detecção de falhas em tempo real, apesar disso, o algoritmo LGR não

foi acelerado pelo arquitetura proposta. Essa decisão se deu por conta dele não possuir custo computacional relevante quando comparado com algoritmo PCA. Evidência verificada na Seção 4.2.1.

Para o terceiro estudo de caso, PCA e LGR não foram utilizados para a detecção de falhas, em vez disso, um modelo simples de monitoramento foi proposto, baseado na comparação de dados estimados por um modelo e os dados mensurados, como apresentado na seguinte seção.

2.5 Detecção de Falhas em Sistemas Desconhecidos

Por meio da estimação do modelo de uma planta de sistema, é possível estimar sua saída em tempo real, permitindo a comparação entre os valores reais de saída da planta e os valores estimados pelo modelo desenvolvido (Marcu *et al.*, 2001) (Figura 2.5).

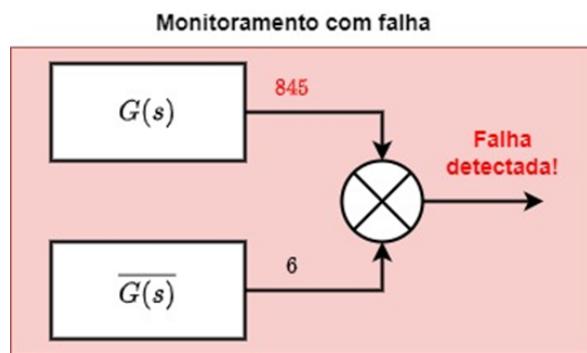


Figura 2.5 – Exemplo genérico de detecção de falhas.

Neste trabalho, a identificação de falhas do primeiro e o segundo estudo de caso utilizaram uma classificação binária (com ou sem falha), através da computação do valor de divergência de cada conjunto de dados comparado com modelos treinados sem nenhuma falha. Ou seja, uma vez que os valores de entrada apresentaram divergência dos dados utilizados no treinamento, uma falha foi detectada. A utilização desses modelos é explicada em detalhes na Seção 2.4 e 2.3.2. Para o terceiro estudo de caso, não foi atribuído um *threshold* de detecção de falhas, nesse caso, foi utilizado o critério estatístico *R-squared* (Teimourzadeh *et al.*, 2021).

Para a modelagem da planta do terceiro estudo de caso, foi escolhido uma rede neural de fácil implementação em *hardware*, alimentada por conjuntos de dados com regressores NARX (Seção seguinte). Uma proposta de aceleração dessa arquitetura é apresentada na Seção 5.2.3.

2.6 NARX em Modelagem de Sistemas

Para permitir que a rede neural proposta capturasse padrões complexos de interação, foi inicialmente utilizado um modelo NARMAX como entrada da rede. Este método é consolidado na literatura e amplamente empregado na identificação de sistemas não lineares, com sucesso comprovado em trabalhos como (Muñoz; Acuña, 2021), (Ahmed *et al.*, 2022) e (Shan; Hou, 2016). Apesar disso, ao tentar utilizar o modelo NARMAX, foram encontradas dificuldades na estimação dos hiperparâmetros relacionados aos termos de erro do sistema – etapa essencial para a modelagem NARMAX. Essa dificuldade decorreu da falta de informações sobre o sistema a ser identificado, caracterizado como um modelo de caixa-preta. Diante disso, o modelo NARX foi escolhido como uma alternativa mais viável, pois não exige a estimação dos termos de erro do sistema. A organização de um conjunto de dados em um modelo NARX é feita através da estrutura apresentada na Equação 2.3.

As entradas do modelo NARX são definidas em termos de entrada $u(t)$ e saída $y(t)$:

$$y(t) = F [y(t-1), y(t-2), \dots, y(t-n_y), u(t-1), u(t-2), \dots, u(t-n_u)], \quad (2.3)$$

onde F é uma função não linear que mapeia as entradas atrasadas para a saída no instante t , n_y representa a quantidade de termos de saída atrasada e n_u representa a quantidade de termos de entrada atrasada. A estimação dos hyper-parâmetros n_y e n_u não é trivial e depende muitas vezes de processos empíricos ou de algoritmos de *brute-force* para encontrar os melhores resultados para uma arquitetura eficiente (Singla; Subbarao; Junkins, 2007).

Neste trabalho, para a função F apresentada na equação 2.3, uma rede neural RBF foi utilizada, por conta de sua simples arquitetura (somente uma camada escondida), por possuir rotinas de treinamento relativamente simples e por ser considerada um aproximador universal de funções, além de apresentar boa capacidade de identificação de sistemas caixa preta (Billings; Voon, 1984).

2.7 Rede Neural RBF

A rede neural de base radial (*Radial Basis Function* - RBF) 2.6 é uma rede neural composta de 3 camadas básicas: Uma camada de entrada, uma camada oculta e uma camada de saída. Na camada oculta da rede, utiliza-se uma função de base radial como função de ativação dos neurônios, normalmente uma função Gaussiana como a apresentada na 2.4.

$$\varphi(\mathbf{x}, \mathbf{c}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}\|^2}{2\sigma^2}\right), \quad (2.4)$$

onde \mathbf{x} representa o vetor de entrada, \mathbf{c} é o centro da base radial e σ é o parâmetro de largura da função, que controla a suavidade da função. O modelo final da rede RBF pode ser expresso como:

$$y(\mathbf{x}) = \sum_{i=1}^N w_i \varphi(\mathbf{x}, \mathbf{c}_i), \quad (2.5)$$

onde $y(\mathbf{x})$ é a saída da rede, N é o número de neurônios na camada oculta, w_i são os pesos da camada de saída e \mathbf{c}_i são os centros de cada função de base radial. Comparativamente, redes RBF possuem baixa complexidade de treinamento quando comparada com redes *deep-learning*, já que o treinamento da rede RBF utiliza de algoritmos com custos computacionais relativamente baixos para a determinação dos parâmetros. Por exemplo, os centros \mathbf{c}_i podem ser estimados pelo método *kmeans* (Tonussi; Quintero, 2020) e os pesos w_i podem ser estimados por *Ordinary-Least-Squares* (Frost, 2019). Apesar da facilidade de treinamento da rede, ela apresenta um bom desempenho para estimativa de sistemas não lineares.

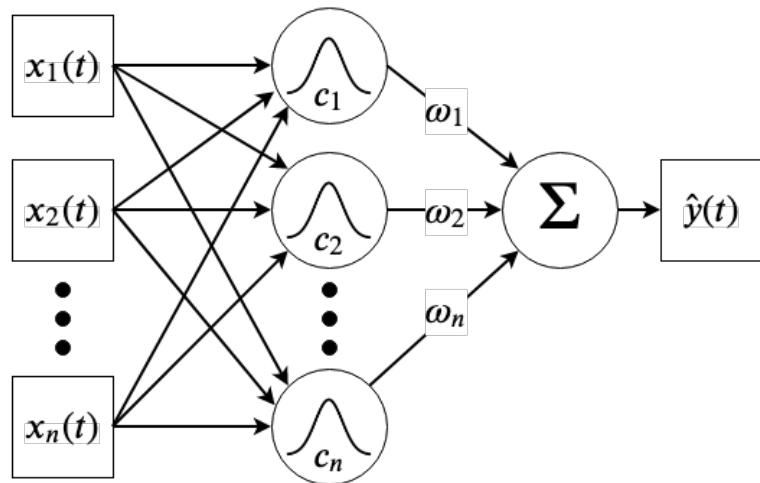


Figura 2.6 – Rede Neural RBF utilizada em (Tonussi; Quintero, 2020).

Visto que não foi feita a classificação binária de falhas por LGR no terceiro estudo de caso, para a validação do modelo proposto, foram utilizados métodos estatísticos clássicos: Validação por *Free-Run* e validação por *One-Step-ahead* (Seção seguinte).

2.8 Métodos de validação utilizados: *Free-Run* e *One-Step-ahead*

No método de validação *one-step-ahead* (OSA), as saídas da planta real são utilizadas para alimentar o próximo passo de estimativa $t + 1$ do modelo identificado (Ljung, 1999). O valor de saída estimado, é então comparado ao valor observado real para avaliar a precisão

do modelo. Se denotarmos a saída real no tempo t como $y(t)$ e a saída prevista pelo modelo como $\hat{y}(t|t - 1)$, o erro da previsão *one-step-ahead* pode ser expressado como na Eq. 2.6:

$$\text{Erro}(t) = y(t) - \hat{y}(t|t - 1). \quad (2.6)$$

Ou seja, para a validação OSA, não existe uma acumulação explícita de erro ao longo do período de validação, pois, o modelo faz sua estimativa somente um passo de tempo por vez. Validar um modelo utilizando somente o OSA pode ser inadequado, já que uma previsão de um único passo à frente pode estar mal feita mas ter seu erro completamente compensado por entradas e saídas da planta real, dando uma falsa indicativo de uma modelagem bem feita.

Diferente do OSA, o método *free-run* (FR) pode servir como uma evidência mais sólida para validação de modelos. O método FR compara a saídas estimadas com as saídas reais de um sistema quando este é deixado para operar livremente, sem a adição de entradas externas provenientes da planta real. A validação por *free-run* é particularmente útil para avaliar o desempenho de um modelo em capturar a dinâmica de um sistema em condições operacionais naturais e para avaliar por quanto tempo um modelo puramente estimado consegue convergir para uma resposta representativa do sistema real (Billings; Voon, 1984).

Se denotarmos a saída do sistema como $y(t)$ e a saída do modelo como $\hat{y}(t)$, o erro da previsão *free-run* pode ser expressado como na Eq. 2.7:

$$\text{Erro}(t) = y(t) - \hat{y}(t). \quad (2.7)$$

Ou seja, diferente do método OSA, erros de estimativa do modelo são acumulados ao longo do período de validação.

Para os treinamentos e validações OSA e FR do terceiro estudo de caso, foi utilizada uma planta real: Um respirador mecânico de baixo custo, desenvolvido pela Universidade de Brasília (UnB), planta explicada em detalhes na Seção seguinte.

2.9 Terceiro Estudo de Caso - Respirador TICÊ

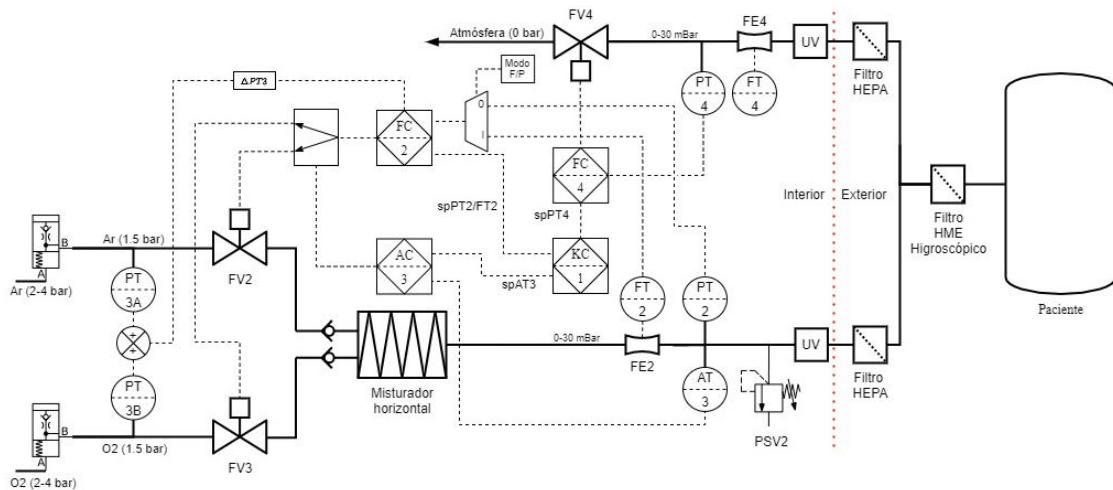


Figura 2.7 – Planta simplificada do respirador mecânico TICÊ (Barbalho, 2021)

O respirador mecânico TICÊ foi a planta utilizada para o terceiro estudo de caso deste trabalho. Em resumo, o sistema do respirador pode ser dividido em 2 partes:

- Parte Interna: Composta de tubulações hidráulicas, atuadores, sensores de pressão e sensores de volume; Encapsulado por uma caixa metálica, com botões de configuração e uma tela *touch* (Figura 2.8).



Figura 2.8 – Caixa do respirador mecânico TICÊ.

- Parte Externa: Fontes de oxigênio e ar comprimido, tubos para interfaceamento invasivo com o paciente e pulmão (orgânico ou artificial), observado na Figura 2.9.



Figura 2.9 – Pulmão artificial conectado com respirador mecânico TICÊ.

Como a proposta neste trabalho é apresentar uma solução não determinística para detecção de falhas do sensor de pressão inspiratória, apenas o sensor de pressão *PT2*, indicado na Figura 2.7, teve sua saída estimada. Apesar disso, os atuadores *FV1*, *FV2*, *FV3*, *FV4* e sensores *FT4*, *AT3*, também foram utilizados, porém, somente como entradas do modelo do sensor sintético.

2.10 Controle do Respirador TICÊ

Para o respirador mecânico desenvolvido (TICÊ), o sistema de controle de oxigênio, pressão e fluxo utiliza um controlador PID profissional, tal como mostrado na Figura 2.10 obtido da tese de doutorado de Bestard, 2017. Esse tipo de PID apresenta diversas vantagens em relação ao controlador PID convencional (BESTARD, 2017), entre elas:

- Proteção *wind-up* no componente integrativo.
- Medições, referência e variável controlada são filtradas dentro do controlador.
- Os valores de medição são utilizados para normalizar os cálculos do controlador.
- O termo integrativo do controlador pode ser inicializado de acordo com o valor inicial do atuador.
- O termo derivativo possui proteção contra "chute" (*kick protection algorithm*).

Para o controle de pressão PEEP (Figura 2.11) é utilizado um sistema de controle com realimentação simples onde o controlador PID2 gera um sinal de controle (*FV4*) para o processo, com base na medição do sensor de pressão após o misturador (*PT4*). Já no controle de pressão e de fluxo utiliza-se um sistema com duas entradas (Figura 2.12): *spPID0* e *spPID1* (referências para os controladores de *pressão* e *concentração de oxigênio*, respectivamente). Pode ser observado que para o controlador *PID 1* o sinal de erro é calculado como a diferença entre a referência de %O₂ e a concentração estimada (através de um preditor Smith). As

saídas são a abertura da válvula de oxigênio (FV2), a abertura da válvula de ar-comprimido (FV3) e a concentração (α).

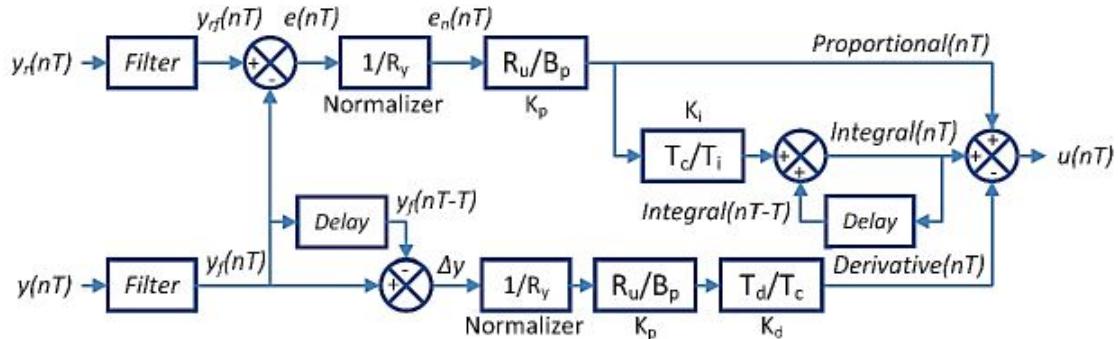


Figura 2.10 – Esquema de controlador PID profissional apresentado em (BESTARD, 2017)

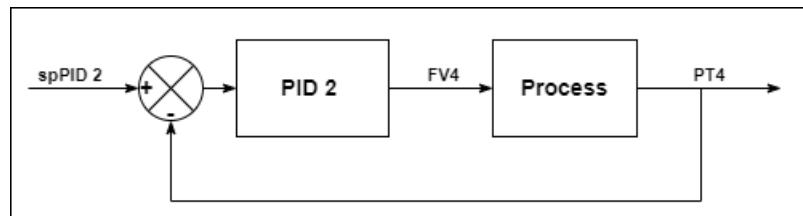


Figura 2.11 – Esquema geral de controle de PEEP.

A saída α da Figura 2.12 é a entrada do preditor Smith que engloba o processo de mistura de oxigênio e ar. O preditor Smith gera como saída o valor da concentração de oxigênio estimada (%O₂, Figura 2.13), eliminando o atraso de transporte de dito sistema de medição.

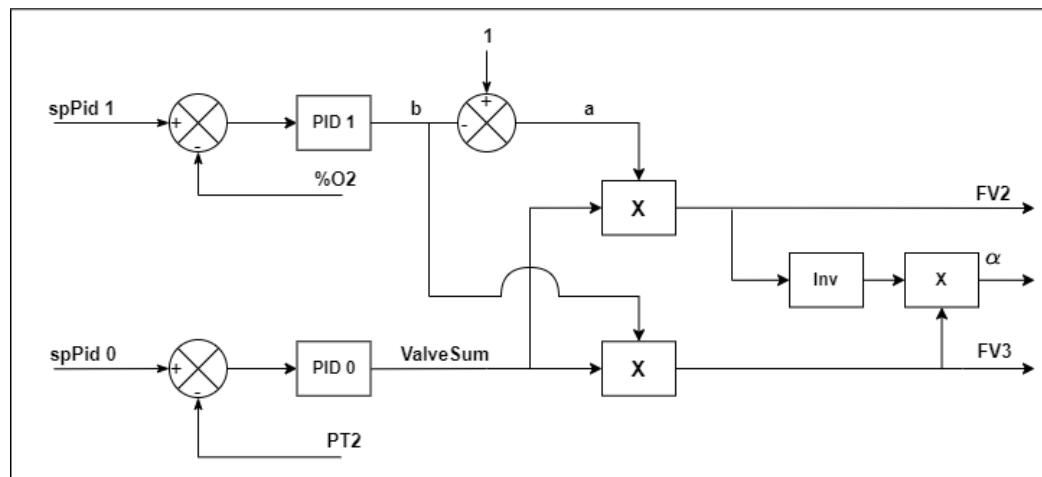


Figura 2.12 – Loop de controle de O₂.

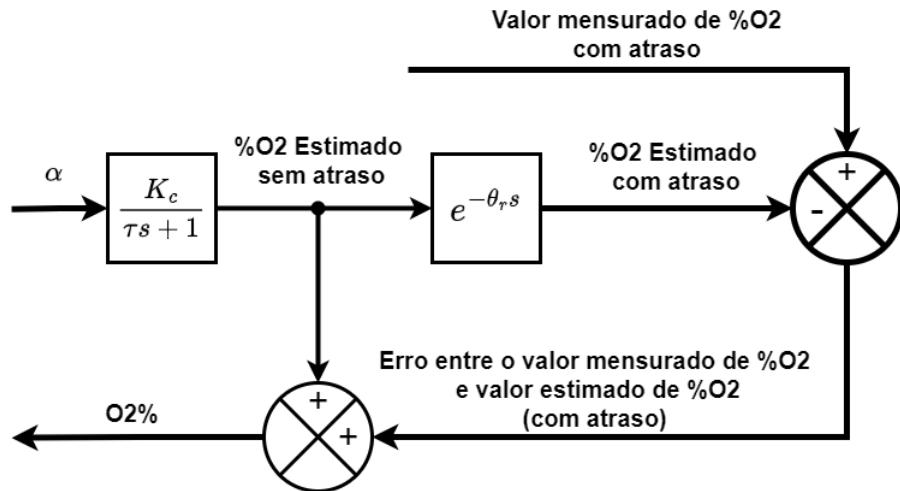


Figura 2.13 – Design do preditor Smith.

A Figura 2.12 define o sistema de controle para %O₂ gerando três saídas correspondentes às saídas das válvulas FV2 e FV3, que controlam três variáveis dependentes entre si (pressão ou volume, %O₂, e a pressão PEEP). As equações 2.8, 2.9, 2.10 e 2.11 mostram como é feito o cálculo das variáveis intermediárias do sistema de controle apresentado.

$$FV2 = a * ValveSum \quad (2.8)$$

$$FV3 = b * ValveSum \quad (2.9)$$

$$a = 1 - b \quad (2.10)$$

$$\alpha = \frac{FV3}{FV2} \quad (2.11)$$

2.11 Considerações finais do capítulo

Neste capítulo, foi apresentada a fundamentação teórica necessária para entendimento do trabalho. Para o desenvolvimento da arquitetura proposta nos dois primeiros estudos de caso, a explicação do funcionamento dos algoritmos implementados foi feita nas Seções 2.3.2 e 2.1. Já para o terceiro estudo de caso (respirador mecânico), outros conceitos foram necessários de se contextualizar: (a) Funcionamento da planta do respirador (b) O modelo NARX utilizado para estimativa de pressão, (c) Os métodos de validação para identificação não determinística, (d) A arquitetura do controlador PID implementado no respirador estudado e (e) A rede neural RBF utilizada para o trabalho de identificação da pressão inspiratória.

3 Revisão da Literatura

3.1 Revisão da Literatura para o Primeiro e o Segundo Estudo de Caso

A partir da análise do estado da arte em estimação de estresse e danos estruturais em materiais, foram identificadas alguns dos métodos mais predominantes na literatura e suas limitações. Desse modo, foi possível selecionar os melhores algoritmos para serem acelerados em uma arquitetura inédita. Por fim, através do estudo do estado da arte, foi possível comprovar a relevância do desenvolvimento de uma nova arquitetura embarcável.

- Estimação de estresse em materiais

Para verificar a importância do primeiro estudo de caso escolhido, foi feito um estudo bibliográfico dos temas de pesquisa relacionados com estimação de estresse por sensores ultrassônicos. Para isso, foi utilizado o software *VOSviewer* e a base de dados *SCOPUS*. A função aplicada no campo de pesquisa de periódicos pode ser observada na Eq. 3.1.

$$(strain \& estimation \& ultrasonic \& material) \quad (3.1)$$

Utilizando a função de agrupamento por co-ocorrência de palavras-chave, foi feita a análise os tópicos mais explorados dentro desse tema (Figura 3.1). Nessa figura, pode ser observado que três *clusters* foram gerados, sendo dois focados em estudos estruturais e um formado de estudos biomédicos, clarificando também, as conexões entre as palavras chaves utilizadas nos artigos. Por exemplo, o tema de (testes ultrassônicos) aparece conectado com (detecção de danos) e com (teste de fadiga).

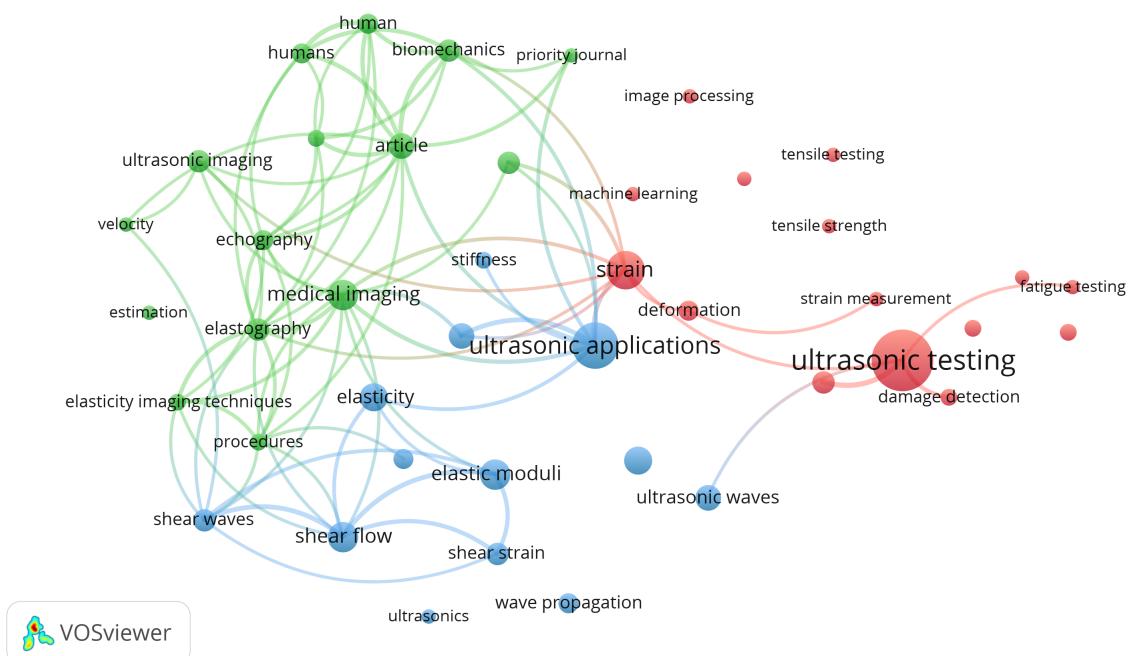


Figura 3.1 – Áreas de pesquisa relacionadas com preditores estimação de estresse + ultrassônico + material.

Após a análise das palavras chaves, foi observado que as áreas mais trabalhadas com estimativa de estresse ultrassônico foram: (a) Estudos envolvendo elasticidade de materiais, (b) Estudos de imagem médicas, (c) Estudos envolvendo detecção de falhas. Isso evidencia uma variação de áreas de estudo que utilizam de estratégias de monitoramento por ondas ultrassônicas, apresentando o interesse da literatura no assunto. Na Tabela 3.1, são apresentados os artigos de maior contribuição no tema, por número de citações, encontrados no estudo bibliográfico da Figura 3.1.

Tabela 3.1 – Artigos sobre estimativa de estresse, selecionados por relevância (número de citações)

REF	Problema	Área de Aplicação	Plataforma computacional	Algoritmo Utilizado
(Chen; Chen; Huang, 2016)	Desenvolvimento de um sistema detecção de deformação cardíaca por ultrassom 3D, otimizado para execução próxima ao tempo real	Engenharia Biomédica	GPUs	<i>Dynamic Programming Method</i>
(Popp <i>et al.</i> , 2020)	Métodos não destrutivos para identificar falhas mecânicas em baterias de lítio por meio de ultrassom	Engenharia Elétrica	Não Apresentada	Time of Flight cross-correlation
(Stanzl-Tschegg <i>et al.</i> , 2016)	Caracterização da superfície de fratura e propagação de trincas pequenas em alumínio	Engenharia Mecânica	Inspeção Manual	Não Aplicado
(Wang; Feng; Jiang, 2017)	Um novo método de avaliação da microplasticidade em fadiga de alto ciclo	Engenharia Mecânica	Não Apresentada	<i>Manson-Coffin</i>
(Wittek <i>et al.</i> , 2016)	Uma abordagem por elementos finitos para identificar propriedades hiperelásticas anisotrópicas de paredes aórticas normais e doentes usando imagens de deformação cardíaca por ultrassom 4D	Engenharia Biomédica	Computador convencional	<i>NELDER-MEAD simplex algorithm</i>

A partir da Tabela 3.1, pode-se verificar que a utilização de técnicas de medição por

ultrassom é um tópico atual e interdisciplinar. Apesar disso, os artigos mais relevantes encontrados, não exploraram conceitos de aceleração de algoritmos, com exceção do trabalho (Chen; Chen; Huang, 2016). Para evidenciar de maneira mais clara como a aceleração de algoritmos é abordada nesse tópico, uma nova pesquisa bibliográfica foi feita, utilizando a função de busca apresentada em Eq. 3.2.

$$(strain \& estimation \& ultrasonic \& material \& optimization) \quad (3.2)$$

A partir dessa pesquisa, somente dois artigos com mais de 2 citações foram encontrados: (Abbas et al., 2021) e (Rivaz et al., 2014).

- **Tecnologias atuais sobre Monitoramento de Dutos**

Para verificar a importância do segundo estudo de caso escolhido, um estudo bibliográfico similar ao primeiro estudo de caso foi feito. Dessa vez, incluindo *optimization* como uma nova palavra-chave na busca. A pesquisa foi novamente feita com o software *VOSviewer* e a base de dados *SCOPUS*. A função aplicada no campo de pesquisa pode ser observada em Eq. 3.3.

$$(pipeline \& ultrasonic \& optimization) \quad (3.3)$$

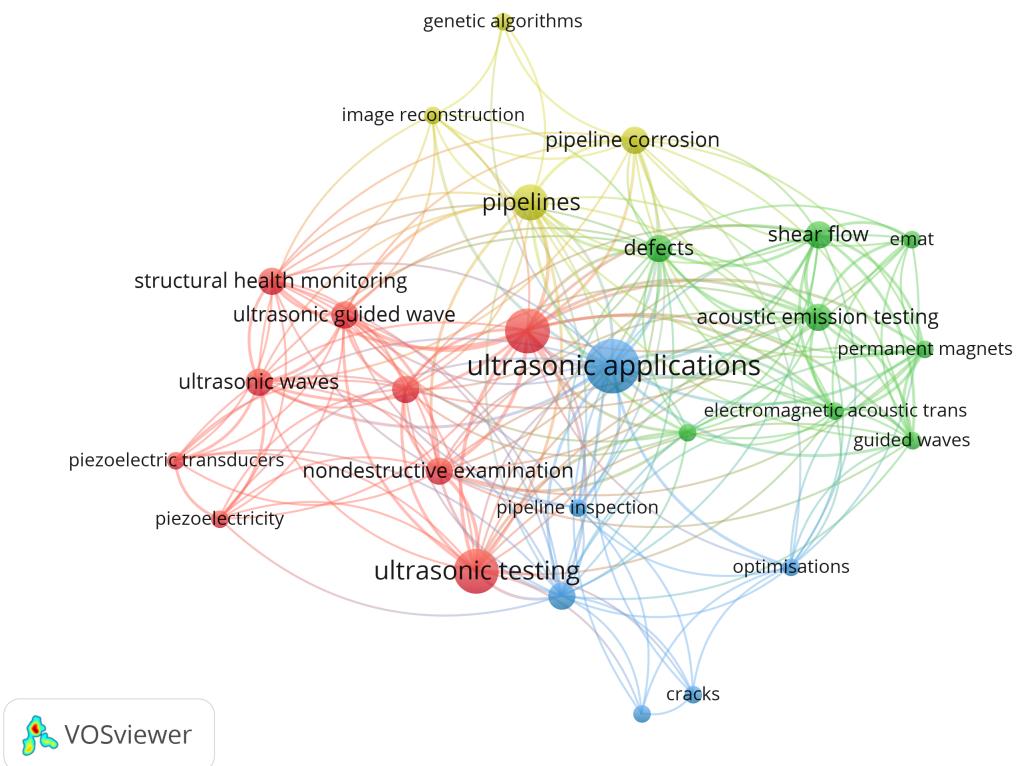


Figura 3.2 – Áreas de pesquisa relacionadas com dutos + ultrassônico + material + otimização.

Após a análise das palavras chaves (Figura 3.2), foi observado que as áreas mais trabalhadas em monitoramento ultrassônico de dutos foram: (a) Testes ultrassônicos, (b) Reconstrução de imagens, (c) Testes acústicos e (d) Otimizações. Apesar de um total de 4 *clusters* principais terem sido formados, as interconexões entre as palavras chave foram abundantes, indicando uma possível concentração de publicações na mesma área de conhecimento. Para verificar essa teoria, foram selecionados os artigos de maior contribuição do assunto Tabela 3.2.

Tabela 3.2 – Artigos sobre monitoramento de dutos, selecionados por relevância (número de citações)

REF	Problema	Área de Aplicação	Plataforma Computacional	Algoritmo Utilizado
(Martin <i>et al.</i> , 2018)	Novo algoritmo para reconstrução de imagens de sensores ultrassônicos	Engenharia Mecânica	Não informado	<i>ECT inversion and TOMOFAST-X</i>
(Park <i>et al.</i> , 2017)	Novo transdutor para monitoramento de dutos e tubulações	Engenharia Mecânica	Não informado	<i>Pulse-echo and through-transmission</i>
(Thon <i>et al.</i> , 2022)	Otimização de um algoritmo para inspeção de tubulações corroídas.	Engenharia Mecânica	GPU	<i>Pogo Finite Element Model</i>
(Niu; Chen; Marques, 2017)	Otimização da matriz de transdutores piezoeletricos através de técnicas de simulação para testes com ondas guiadas em tubulações	Engenharia Mecânica	Computador Convencional (MATLAB)	<i>Finite Element Analysis</i>

Novamente, assim como na análise bibliográfica do primeiro estudo de caso, os artigos com maiores números de citações não exploraram otimizações em *hardware* dedicado.

3.1.1 Revisão Geral da bibliografia de arquiteturas para detecção de falhas

Com o objetivo de melhor entender os trabalhos mais relevantes relacionados a otimizações de algoritmos para monitoramento de estruturas, foi realizada uma nova pesquisa bibliográfica. Nessa pesquisa final, três critérios objetivos para classificar os artigos foram escolhidos:

- Algoritmo utilizado para monitoramento: Verifica qual algoritmo foi implementado para o monitoramento da estrutura.
- Estratégias para redução de conjunto de dados: Especifica quais estratégias foram empregadas para a redução do conjunto de dados.
- Se o algoritmo é embarcado: Uma simples verificação se o algoritmo de monitoramento proposto foi embarcado.

Para filtrar os artigos, alguns critérios subjetivos foram definidos: Relevância para o tema, citações e data de publicação; nesta ordem de prioridade. Os resultados podem ser vistos na 3.3.

Tabela 3.3 – Resultados da pesquisa bibliográfica para a arquitetura implementada

REF	Resumo	Algoritmo utilizado	Estratégias para redução dos conjuntos de dados	O algoritmo é embarcado?
(Ying <i>et al.</i> , 2013)	Detectão de danos em tubulações de aço, através de técnicas de monitoramento estrutural	<i>SVMs e AdaBoost</i>	Normalização por energia e filtragem passa-baixa	Não
(Sen <i>et al.</i> , 2019)	Detectão de danos em tubulações, utilizando algoritmos de aprendizado de máquina	<i>multinomial logistic regression</i>	PCA	Não
(Lee <i>et al.</i> , 2013)	Previsão de falhas em oleodutos e gasodutos utilizando monitoramento em tempo real	<i>Euclidean-SVM</i>	Nenhuma Aplicada	Não
(Dehghan-Niri; Salamone, 2015)	Nova método de monitoramento estrutural e identificação de corrosão	<i>probabilistic reconstruction algorithm</i>	Nenhuma Aplicada	Não
(Xu <i>et al.</i> , 2019)	Detectão de defeitos em tubulações utilizando transdutores PZT	<i>Finite element method (FEM)</i>	<i>matching pursuit (MP)</i>	Não
(Bouzenad <i>et al.</i> , 2019)	Monitoramento da integridade de tubulações por método <i>K-means</i>	<i>K-means</i>	Nenhuma Aplicada	Não
(Gao <i>et al.</i> , 2015)	Monitoramento da quantidade de areia em reservatórios através de ondas ultrassônicas	<i>Deterministic Model</i>	<i>Wavelet decomposition</i>	Não
(Brath <i>et al.</i> , 2017)	Detectão de corrosão em curvas de tubulações	<i>INELAN</i>	Média de sinais de entrada	Não
(Livadiotis; Ebrahimkhanlou; Salamone, 2019)	Monitoramento de tubulações através de reconstrução algébrica	<i>Algebraic reconstruction technique (ART)</i>	Nenhuma Aplicada	Não
(Dubuc; Ebrahimkhanlou; Salamone, 2018)	Algoritmo para localização de múltiplos estímulos de ultrassom em tubulações	<i>matching pursuit (MP)</i>	Nenhuma Aplicada	Não

A pesquisa bibliográfica apresentada na Tabela 3.3 permite visualizar quais algoritmos são mais abordados na literatura para o monitoramento de dutos por ultrassom. Entre esses artigos, os trabalhos (Sen *et al.*, 2019), (Xu *et al.*, 2019) e (Gao *et al.*, 2015) se destacam pela utilização de algoritmos complexos para a redução do conjunto de dados estudado. Desses artigos, (Sen *et al.*, 2019) foi classificado como o segundo artigo de maior importância, de acordo com o critério escolhido, utilizando o algoritmo PCA para a redução de dimensionalidade do modelo. A partir da mesma Tabela, também foi possível perceber o baixo interesse dos artigos com maior importância na aplicação de métodos de monitoramento em plataformas computacionais embarcadas.

3.2 Revisão da Literatura para o Terceiro Estudo de Caso

3.2.1 Sobre Controle e Correção de Falhas em Ventiladores Mecânicos

Para verificar a importância do terceiro estudo de caso escolhido, foi feito um estudo bibliográfico dos temas de pesquisa relacionados com controle de respiradores mecânicos. Para isso, foi utilizado o software *VOSviewer* e a base de dados *SCOPUS*. A função aplicada no campo de pesquisa de periódicos pode ser observada na Eq. 3.4.

$$(\text{mechanical} \& \text{ventilator} \& \text{control}) \& (\text{subarea, "Engineering"}) \quad (3.4)$$

Como pode ser observado na Figura 3.3, essa pesquisa gerou 3 clusters com alto grau de interconexão. Ou seja, a maioria dos artigos publicados nessa área abordam temas similares. Por exemplo, muitos dos artigos observados nessa pesquisa bibliográfica exploraram os temas: Covid-19, tratamento intensivo (UTI), simulação computacional e identificação de sistema de controle.

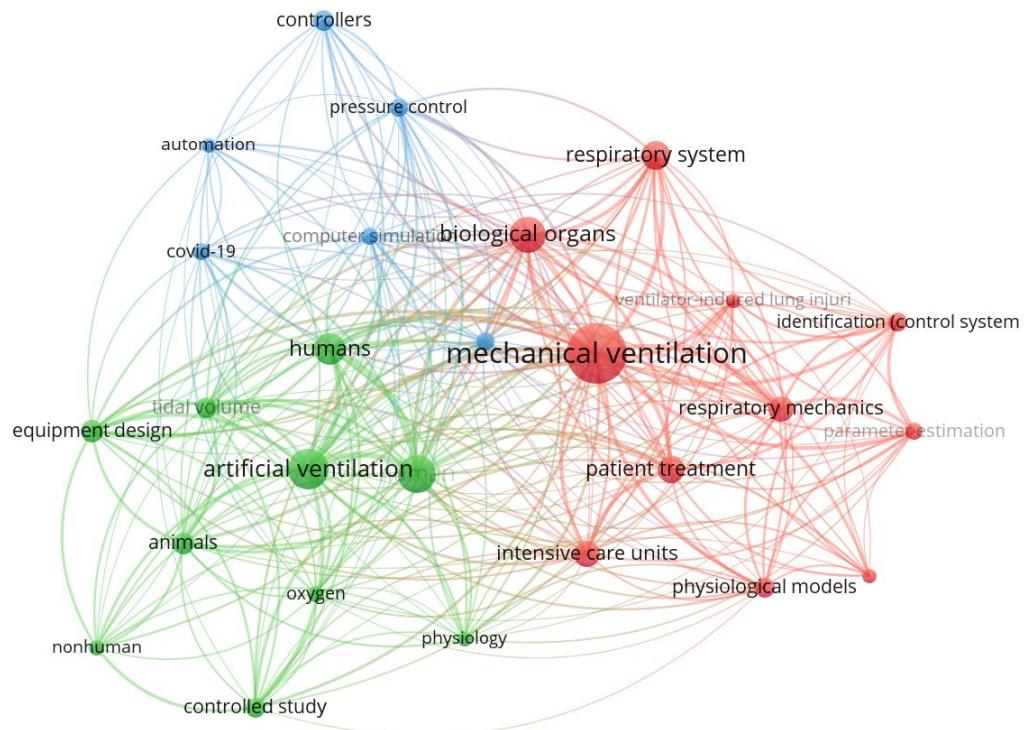


Figura 3.3 – Áreas de pesquisa relacionadas ao controle de ventiladores mecânicos.

Como foram obtidos muitos resultados com o filtro 3.4 (cerca de 200 artigos), foi utilizado um critério de filtragem para selecionar as pesquisas de inovação com os maiores números de citações (Tabela 3.4). Além disso, também foi feita uma classificação desses trabalhos em 4 categorias diferentes: Problema, estratégias de controle, estratégias de correção de erro e testes em seres vivos.

Tabela 3.4 – Trabalhos principais relacionados com respiração mecânica com alto índice de citação.

REF	Problema	Estratégias de Controle	Estratégias de Correção de Erro em Medições	Testado em seres vivos?
Hernandez, 2016 (Hernandez <i>et al.</i> , 2016)	Desenvolvimento de sensor de umidade para ventiladores mecânicos	Não aplicável	Ajuste por mínimos quadrados	Sim
Robert, 2010 (Robert <i>et al.</i> , 2010)	Desenvolvimento de um novo regulador para ventiladores mecânicos	Controlador de pressão com <i>lag</i> de fase	Nenhum	Sim
Sundaresan, 2011 (Ashwath <i>et al.</i> , 2011)	Novo método para otimizar valores de PEEP em pacientes ARDS	Nenhum	Linearização e normalização de medições	Sim
Beaulieu2012 (Beaulieu <i>et al.</i> , 2012)	Identificação do sistema de impedância de um ventilador mecânico	Controlador com <i>feedback</i> proporcional	<i>Welch's overlapped-segment averaging</i> (WOSA), Função de desidade spectral cruzada média.	Sim
Massaroni2017 (Massaroni <i>et al.</i> , 2017)	Sistema de medição para monitoramento de temperatura e umidade em ventiladores mecânicos	Nenhum	Filtro Notch	Não

Pode-se concluir dessa pesquisa que muitos dos trabalhos não utilizaram estratégias simultâneas para redução de erro em medições e estruturas de controle.

3.2.2 Revisão Geral da bibliografia do terceiro estudo de caso

Com o objetivo de melhor entender os trabalhos mais relevantes relacionados ao controle de ventilação mecânica, foi realizada uma nova extensa pesquisa bibliográfica. Nessa pesquisa final, três critérios objetivos para classificar os artigos foram escolhidos:

- Estratégias de controle: Descreve quais técnicas e/ou controladores comuns foram utilizados.
- Estratégias de redução de erro: Descreve quais estratégias de filtragem ou de detecção de falhas foram utilizadas nos dados antes de alimentá-los no sistema de controle.

- Testados em humanos/animais: Uma simples verificação para confirmar se os testes foram realizados em seres vivos.

Para filtrar os artigos, foram definidos critérios subjetivos, na seguinte ordem de prioridade: Relevância para o tema, citações e data de publicação. Os resultados podem ser vistos na Tabela 3.5.

Tabela 3.5 – Resultados da pesquisa bibliográfica para o respirador mecânico. (1/2)

REF	Resumo	Estratégias de controle	Estratégias de redução de erro	Testado em humanos/animais vivos?
Hernandez2016 (Hernandez et al., 2016)	Um novo tipo de sensor de umidade é desenvolvido, testado e validado em experimentos reais. O sensor é testado dentro de uma linha inspiratória de um ventilador mecânico.	Não Aplicável	Regressão por mínimos quadrados	Sim
Robert2010 (Robert et al., 2010)	Um novo ventilador mecânico funcional é desenvolvido usando uma tecnologia de inspiração líquida. O artigo também explica como técnicas de controle simples podem ser usadas nesse novo sistema. Testes são feitos em simulações de computador, pulmões artificiais e animais. O controle é feito via ventilação controlada por pressão (PCV).	Controlador de atraso de fase	Não	Sim
Sundaresan2011 (Ashwath et al., 2011)	Um novo modelo matemático é desenvolvido capaz de calcular o valor correto de PEEP para pacientes com síndrome do desconforto respiratório agudo (SDRA). Este modelo usa pressões de fechamento e abertura, volume aplicado e capacidade pulmonar total.	Não	Linearização e normalização de medições	Sim
Beaulieu2012 (Beaulieu et al., 2012)	Uma nova estratégia de estimulação volumétrica é desenvolvida para ser aplicada no canal inspiratório de um ventilador mecânico. Para isso, é criado um novo modelo que descreve como o fluxo de entrada e a pressão estão relacionados.	Controle de feed-back proporcional	Média segmentada sobreposta de Welch e função de densidade espectral cruzada média	Sim
Massaroni2017 (Massaroni et al., 2017)	Um novo sistema é desenvolvido para medir simultaneamente a umidade e a temperatura em uma linha de fluido de um ventilador mecânico.	Não	Filtro Notch	Não
Chatburn2011 (Chatburn; Mireles-Cabodevila, 2011)	Diferentes modos de ventilação são estudados, analisando cuidadosamente o sistema de controle de cada um.	Não Aplicável	Não Aplicável	Não Aplicável
Luepschen2007 (Luepschen; Zhu; Leonhardt, 2007)	Uma nova estratégia de recuperação de paciente é proposta para falha respiratória aguda. Esta estratégia usa um controle PID-Smith aplicado a um ventilador mecânico comercial.	Controlador PID-Smith	Não	Sim
Mesic2003 (Mešić et al., 2003)	Um novo sistema mecânico é desenvolvido para simular o comportamento pulmonar humano. O sistema é testado e validado em um protótipo real.	Controlador PID	Linearização em pontos operacionais específicos	Não Aplicável
Tehrani2013 (Tehrani, 2013)	Uma nova estratégia de controle é desenvolvida com base na ventilação assistida proporcional (PAV), juntamente com sua implementação em hardware. Um controlador especializado é desenvolvido para esse propósito.	Controlador não especificado	Filtro de média móvel simples e "Ventilação Média Medida"(AMMV)	Não
Lei2017 (Lei, 2017)	Diferentes tipos de ventilação mecânica são estudados, analisando as variáveis importantes para cada tipo de controle.	Não Aplicável	Não Aplicável	Não Aplicável
Tehrani2008-1 (Tehrani, 2008a)	Um estudo bibliográfico é feito sobre diferentes tipos de técnicas de controle para ventilação mecânica.	Não Aplicável	Não Aplicável	Não Aplicável

Tabela 3.6 – Resultados da pesquisa bibliográfica para o respirador mecânico. (2/2)

REF	Resumo	Estratégias de controle	Estratégias de redução de erro	Testado em humanos/animais vivos?
Tehrani2008-2 (Tehrani, 2008b)	Estudo bibliográfico sobre diferentes tipos de técnicas modernas de controle para ventilação mecânica.	Não Aplicável	Não Aplicável	Não Aplicável
Tehrani2004 (Tehrani et al., 2004)	Desenvolvimento de uma nova estratégia de controle baseada na combinação de duas outras técnicas patenteadas.	Controlador PID com algoritmo "step-wise"	Não	Sim
Smith1996 (Smith; Shneerson, 1996)	Estudo comparativo usando 4 ventiladores mecânicos comerciais em pulmão humano.	Não Aplicável	Não Aplicável	Não
Sayin2018 (Sayin, 2018)	Desenvolvimento de um novo ventilador mecânico para facilitar o transporte de bebês com falência respiratória. Modelos de simulação e um protótipo são criados.	Controlador PID	Filtro de média móvel simples	Não
Mushin1954 (Mushin, 1954)	Estudo bibliográfico sobre diferentes tecnologias disponíveis em ventiladores mecânicos (1954).	Não Aplicável	Não Aplicável	Não Aplicável
Vishnoi1991 (Karab,)	Desenvolvimento de um novo ventilador mecânico capaz de administrar anestésicos para pacientes.	Controlador adaptativo com horizonte rececente	Estimação de variáveis usando mínimos quadrados com fator de esquecimento	Sim
Radzi2021 (Radzi, 2021)	Proposta de uma nova estratégia de controle para equilibrar a mistura de ar e O2 em um ventilador mecânico utilizado em UTIs.	Controlador PID	Não	Não
Borrellido2005 (Borrello; Enterprises; Diego, 2005)	Estudo bibliográfico sobre diferentes modelos matemáticos necessários para o desenvolvimento de ventiladores mecânicos.	Não Aplicável	Não Aplicável	Não Aplicável
Id2020 (Id et al., 2020)	Desenvolvimento de um ventilador mecânico de produção rápida e não patenteado para ajudar no tratamento de pacientes contaminados com COVID-19.	Controle puramente mecânico	Não	Sim
Rogers2003 (Rogers et al., 2003)	Desenvolvimento de uma estratégia de controle para a mistura de ar e O2 em um ventilador mecânico.	PID	Não	Sim
Kiratiwudhikul2014 (Kiratiwudhikul; Chanya-gorn, 2014)	Proposta de uma nova estratégia de controle para equilibrar a mistura de ar e O2 em um ventilador mecânico utilizado em oxigenoterapia. FiO2 é determinado por LUTs via web.	Não	Não	Não
Rodrigues2022 (Rodrigues, 2022)	Estudo bibliográfico sobre diferentes estratégias de equilíbrio da mistura O2-Ar.	Não Aplicável	Não Aplicável	Não Aplicável

A pesquisa bibliográfica apresentada na Tabela 3.5/3.6 proporcionou uma visão mais clara sobre quais estratégias de controle estão sendo mais exploradas neste tema. Os artigos (Luepschen; Zhu; Leonhardt, 2007), (Mešić *et al.*, 2003), (Tehrani *et al.*, 2004), (Sayin, 2018), (Radzi, 2021) e (Rogers *et al.*, 2003) utilizam controladores PID com diferentes níveis de complexidade, sendo (Luepschen; Zhu; Leonhardt, 2007) o único a usar um controlador PID-Smith. Métodos mais simples também foram utilizados, como o controlador de atraso de fase simples em (Robert *et al.*, 2010), o controlador proporcional de *feedback* de (Beaulieu *et al.*, 2012) e o controle mecânico simples utilizado em (Id *et al.*, 2020). Uma observação interessante é que apenas o artigo (Tehrani *et al.*, 2004) utilizou uma versão mais moderna do simples controle PID. O respirador mecânico proposto neste projeto, por exemplo, utiliza uma arquitetura Professional-PID, explicada em detalhes na Seção 2.10.

3.3 Considerações finais do capítulo

A pesquisa bibliográfica apresentada neste capítulo indica uma alta relevância no tópico de monitoramento de materiais por meio de ondas ultrassônicas, estratégia de sensoriamento utilizado no primeiro e no segundo estudo de caso. Contudo, poucos trabalhos exploraram otimizações nos algoritmos utilizados. Buscando preencher essa lacuna, nas Seções 4.1 e 4.2 é apresentada uma solução embarcável para os métodos de estimativa por monitoramento de ondas ultrassônicas, com foco na aceleração da etapa de inferência do algoritmo de redução de dimensionalidade PCA, explicado em detalhes na Seção 2.3.2. Trata-se de uma solução concreta, mas ainda pouco explorada de forma evidente na literatura.

No estudo bibliográfico apresentado sobre o terceiro estudo de caso, apenas (Tehrani *et al.*, 2004) empregou uma versão moderna do controle PID tradicional, porém, sem utilizar estratégias de detecção e falhas. Em geral, existem diversos trabalhos na área de respiração mecânica, apesar disso, poucos desses estudos exploram estratégias de controle modernas ou inéditas. Neste trabalho, é apresentado uma proposta de monitoramento de falhas que pode ser acoplada ao controlador do respirador mecânico, permitindo uma maior integridade dos sensores utilizados na malha de controle.

4 Metodologia

4.1 Arquitetura Desenvolvida para o Primeiro Estudo de Caso

O IP-Core HDL proposto implementa uma arquitetura parametrizada em árvore para executar operações de multiplicação e soma capaz de calcular o produto escalar entre dois vetores (*vector multiplication module*). Essa arquitetura utiliza um registrador sequencial Serial-Input Parallel-Output (SIPO), que armazena os dados de entrada do algoritmo e os componentes principais (PCs) descritos na Seção 2.3.2.

- No lado PL, a arquitetura se conecta com um módulo *Direct Memory Access* (DMA) utilizando uma interface AXI-Stream com entrada e saída e um *bus* de 32 bits. A Figura 4.1 apresenta o esquema de interface de alto nível proposto.
- No lado PS, um processador Zynq interage com o DMA através de portas AXI4-Full de alto desempenho para transferência de memória e utiliza de uma interface AXI-Lite simples para o controle de fluxo. Em resumo, a conexão AXI-DMA oferece uma conexão direta entre o registrador SIPO no HDL-IP Core e um banco de memória endereçável no PS (ARM).

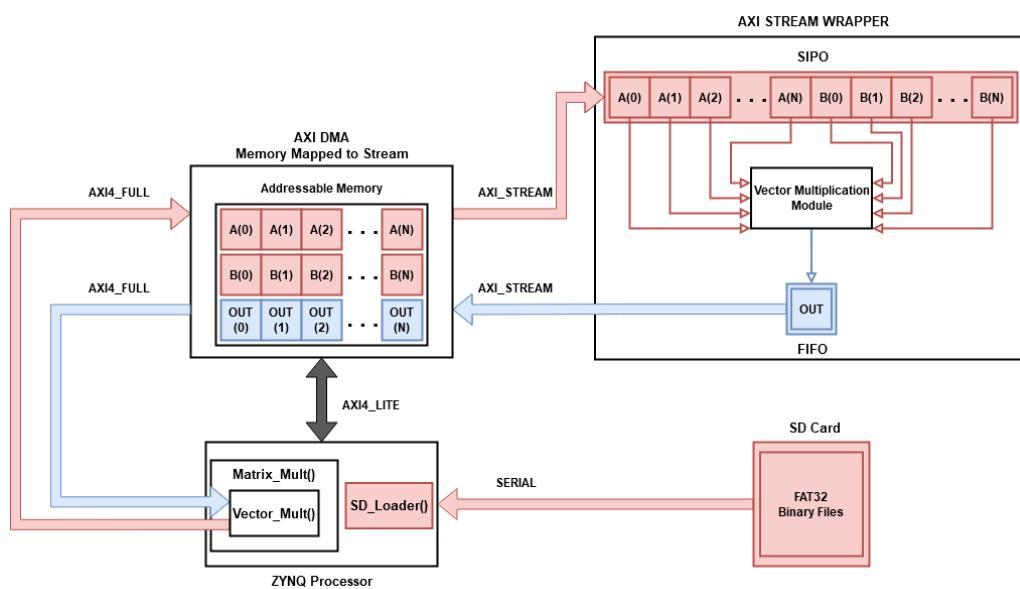


Figura 4.1 – Esquema de interface do IP-Core HDL proposto. Apresentado em (Mendes et al., 2023a)

Quanto ao fluxo de dados dentro do IP-Core, o registrador SIPO carrega as entradas de dados enquanto a multiplicação vetorial é realizada, permitindo o carregamento de novos dados de entrada no registrador SIPO em modo pipeline. Uma máquina de estados finitos controla as diversas etapas do cálculo:

1. Carregamento de dados;

2. Multiplicação de dados;
3. Soma das saídas dos multiplicadores;
4. Envio do resultado.

A Figura 4.2 detalha o módulo implementado no PL. O uso de um registrador de entrada SIPO e um registrador de saída FIFO facilita o desenvolvimento da lógica adicional necessária para o protocolo de *handshake* do AXI-stream.

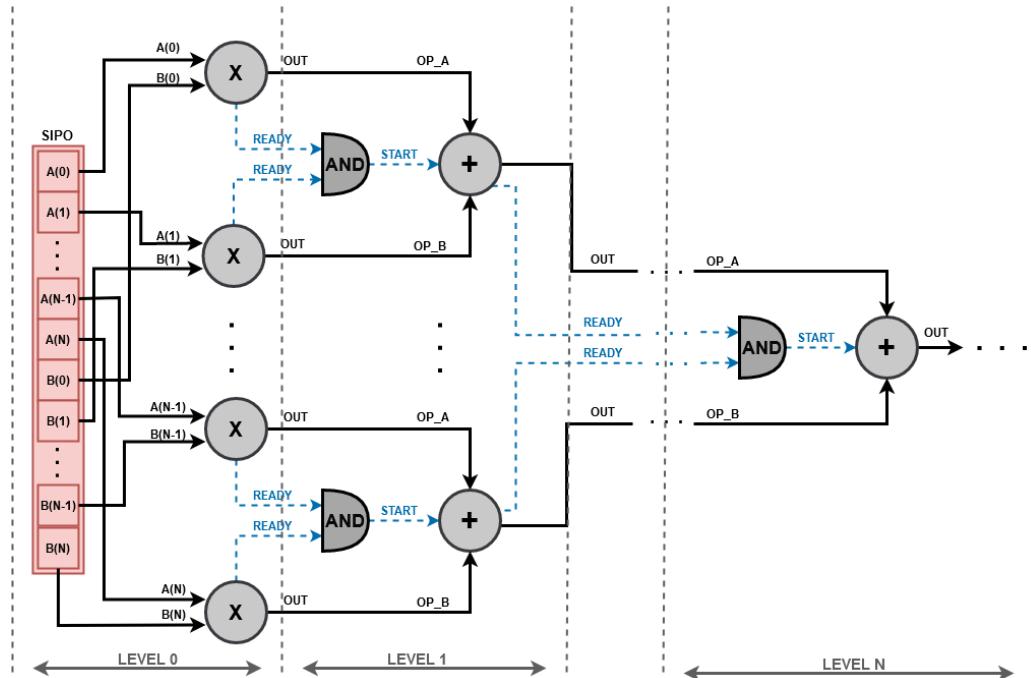


Figura 4.2 – O *Vector Multiplication Module* (representado no bloco *AXI Stream Wrapper* na Fig. 4.1). Apresentado em (Mendes *et al.*, 2023a)

Os operadores de ponto flutuante utilizados nos multiplicadores e somadores são baseados nos IP-Cores pré-caracterizados FPLib, desenvolvidos por (Muñoz *et al.*, 2010a; Muñoz *et al.*, 2010b). O FPLib suporta representações paramétricas de largura de bits customizada em ponto flutuante, facilitando análises de consumo de hardware, desempenho computacional e precisão numérica (Muñoz *et al.*, 2010b).

É possível configurar a quantidade de operadores paralelos de multiplicação/soma na arquitetura em árvore, conforme as necessidades da aplicação. Neste trabalho, foram consideradas duas variações principais: uma com 5 *cores* e outra com 100 *cores*, estruturadas com 6 e 7 níveis de profundidade, respectivamente.

O parâmetro de 5 *cores* foi selecionado por conta de sua latência corresponder ao tempo necessário para preencher o buffer de entrada e processar a multiplicação vetorial. Ou seja, o tempo de preenchimento da SIPO é o mesmo do tempo de execução do resto da lógica. Já o valor de 100 *cores* foi escolhido como o limite máximo viável para implementação no FPGA utilizando operadores de ponto flutuante de precisão simples. Cada estágio de soma é

conectado ao estágio anterior de multiplicação através de uma sincronização feita por portas AND, ligando o sinal de ready dos multiplicadores com o sinal start do somador.

A divisão de tarefas entre PS e PL foi realizada da seguinte forma: O componente PL é responsável por multiplicar uma seção da multiplicação matricial, enquanto o PS é responsável por fazer o *streaming* de dados para o PL, gerenciar interrupções criadas pela arquitetura, acumular resultados parciais no resultado final e alinhar as seções de memória para transmissão à ponte AXI-DMA. Essa distribuição de tarefas é mostrada na Eq. 4.1.

$$C_{ij} = \underbrace{\sum_1^M}_{\text{PS}} \underbrace{\sum_k^{cores+k} A_{ik} \cdot B_{kj}}_{\text{PL}}, \quad (4.1)$$

em que M é a quantidade necessária de chamadas à arquitetura implementada, calculada como $M = \lceil 8192/cores \rceil$. Isso implica que, para a implementação com 100 cores, são necessárias 82 chamadas para obter um único resultado, enquanto para 5 cores são necessárias 1640 chamadas. Além disso, o conjunto de dados e os componentes principais necessários foram armazenados em um cartão SD externo, sendo carregados de acordo com a necessidade.

Toda a arquitetura apresentada é executada em clock de 100 Mhz para o PL e 666Mhz para o PS. Os valores de clock escolhidos foram os valores máximos comportáveis pelo *design*. Valores de clock acima de 100Mhz não conseguiram atender os limites de *timing* da implementação real em FPGA.

4.2 Melhorias da Arquitetura Para Sua Utilização no Segundo Estudo de Caso

Para o estudo de caso de monitoramento de dutos, algumas melhorias na arquitetura foram implementadas visando maximizar a eficiência computacional se comparado com o estudo de caso da Seção 5.1.1. As principais melhorias foram:

- Eliminação da máquina de estados do módulo de multiplicação vetorial apresentado na Figura 4.2. Ao remover a máquina de estados, um pipeline completo foi possível de ser implementado, com cada estado sendo executado em 2 ciclos de clock, como mostrado na Figura 4.3.

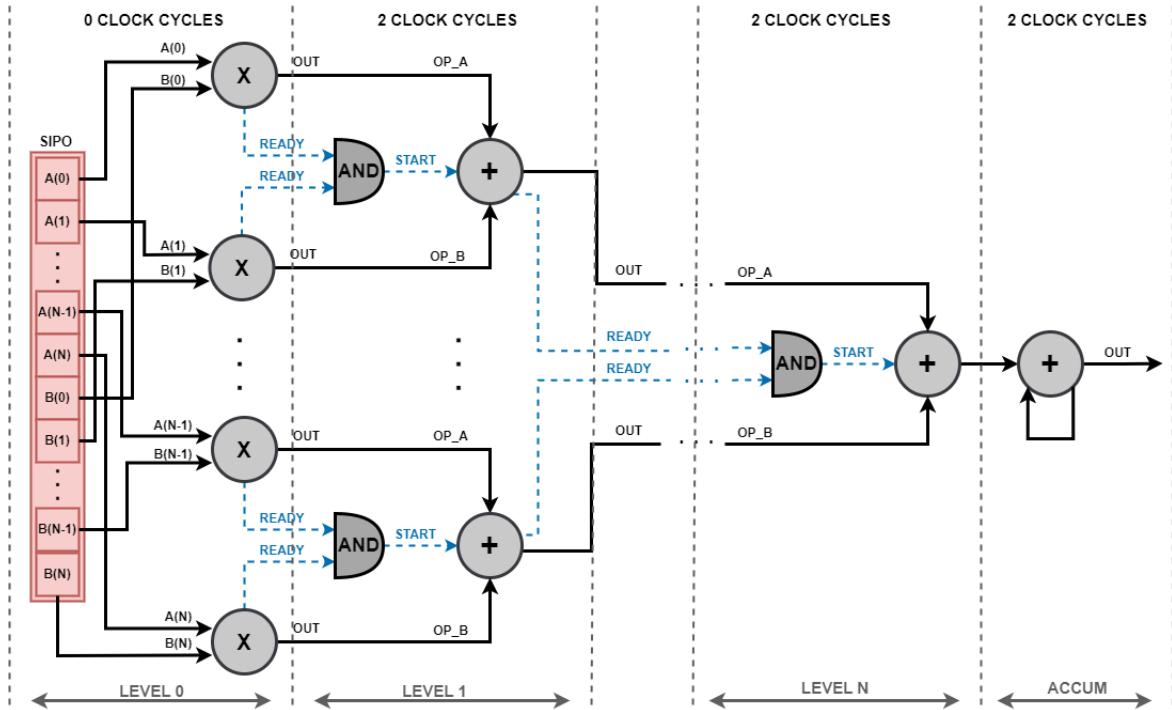


Figura 4.3 – Bloco de multiplicação vetorial com *pipeline* completo.

- Realocação da acumulação das multiplicações vetoriais do ARM para FPGA: Relocar a acumulação dos resultados de multiplicação vetorial do código em ARM para a FPGA diminuiu significativamente a necessidade de transferência de dados entre sistemas. A saída do IP na FPGA só é transmitida quando a multiplicação de todos os vetores necessários para compor um dado da matriz resultado é finalizada.
- Mudança na comunicação entre AXI DMA e o *wrapper Axi4-Stream*: A remoção da máquina de estados do sistema permitiu que o envio de dados de maneira contínua e ininterrupta. Através do protocolo AXI-Stream DMA é possível enviar um dado a cada ciclo de clock (vide Figura 4.4).

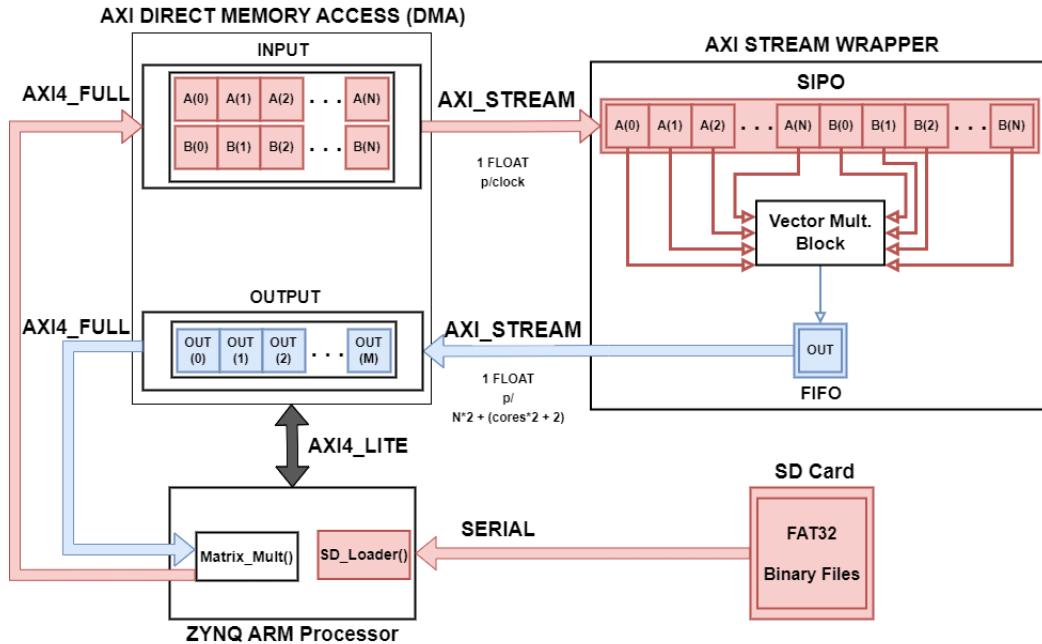


Figura 4.4 – Diagrama de alto nível para a arquitetura proposta para monitoramento de dutos.

- A lógica foi modificada no código em PS para permitir o envio de dados entre ARM e DMA de maneira mais eficiente: Para todos os dados transmitidos via DMA, a confirmação de que a transmissão foi completada não é aguardada desde o início da transmissão, isto é, o sinal TxDone não bloqueia o envio contínuo de dados, sendo necessário aguardar tal confirmação apenas quando o último dado da matriz de entrada é enviado (vide Código 4.1).

Código 4.1 – Mudança implementada na comunicação Tx-Rx entre ARM e DMA

```

1   if (groups==groups_max_low){ //se for o ultimo dado
2       SendAndWait(TxBufferPtr ,RxBufferPtr); //aguarda fim da
3       transferencia
4   }else{
5       SendAndForget(TxBufferPtr ,RxBufferPtr); //envia dados de
6       maneira sequencial
7   }

```

4.2.1 Implementação puramente em SW (ARM) e *code-profiling*

Para fins de comparação de performance, também foi desenvolvida uma aplicação em C para execução do algoritmo puramente em *software*, aproveitando uma arquitetura *dual-core* de um processador ARM.

O modelo LGR e o PCA foram implementados em C no processador ARM Cortex-A9 integrado à placa Zybo Z7-20, sendo testados em configurações de um núcleo e dois núcleos. Para fins de comparação, a mesma implementação foi executada em um processador Intel i5

(1035G7). É importante destacar que todos os algoritmos implementados foram aplicados somente após os coeficientes do LGR terem sido calculados.

Para identificar as tarefas de maior custo computacional do algoritmo, foi realizado um perfilamento do PCA-LGR utilizando a IDE *DevCpp*. Os resultados mostraram que 98,5% do tempo total de processamento foi gasto na etapa de multiplicação de matrizes do PCA, enquanto o carregamento de dados representou 0,7%, e a inferência do modelo consumiu apenas 0,8%. Dado que a multiplicação de matrizes foi a etapa mais demorada, essa operação foi paralelizada entre os núcleos do ARM. Uma estratégia simples de paralelismo dividiu a computação igualmente entre os dois núcleos: o primeiro processou metade da multiplicação de matrizes, enquanto o segundo foi responsável por processar a outra metade.

As etapas implementadas no processador ARM foram:

1. Carregar a imagem de teste do cartão SD para a RAM (matriz X);
2. Calcular a média de cada coluna do conjunto de dados X ;
3. Subtrair as médias da matriz X ;
4. Realizar a multiplicação de matrizes com os componentes principais (calculados durante o treinamento) para metade dos dados em cada núcleo;
5. Executar a inferência do modelo (produto escalar).

O núcleo 1 processa metade da quarta etapa, enquanto o núcleo 2 processa a outra metade. Foram criadas duas aplicações separadas para cada núcleo com dados e códigos independentes. Como não era possível compartilhar variáveis entre as aplicações, foi definida uma região de memória compartilhada nos *scripts* do *linker* do processador ARM, permitindo que ambos os núcleos utilizassem a mesma área de memória *heap*, embora mantivessem *stacks* separadas. Assim, o cartão SD precisa carregar os dados apenas uma vez na área de *heap* compartilhada.

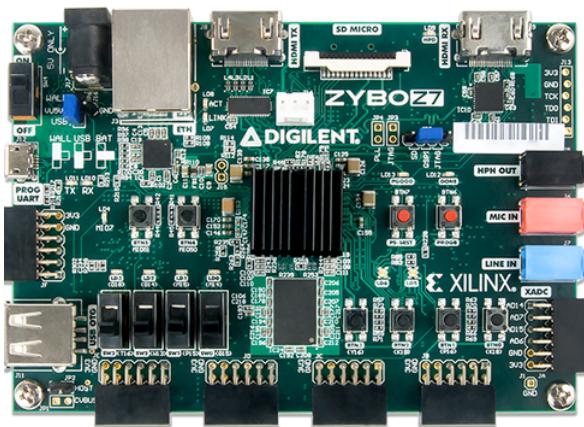
Nessa abordagem baseada em *software*, as imagens B-scan foram armazenadas no cartão SD utilizando o sistema de arquivos FAT32, implementado através da biblioteca FAT32 da Xilinx. Para interfacear a arquitetura com um computador, foi utilizada a comunicação UART-USB.

4.2.2 Plataformas Computacionais usadas para Avaliação de Desempenho

Para avaliar o desempenho computacional do algoritmo em plataformas *off-the-shelf*, os códigos foram executados em um computador comercial Intel, em um processador ARM e na arquitetura proposta em SoC FPGA:

- Intel i5 (1035G7): Executado através de um código compilado em C, rodando em um sistema operacional Windows.

- ARM Cortex-A9 : Executado através de um código compilado em C, rodando em um sistema *bare-metal*. Para a ARM, o código foi testado em *dual-core* (permitindo um nível paralelização de tasks) e em *single-core* (sem paralelização de tasks).
- SoC FPGA Zybo Z-20: para a implementação da arquitetura de hardware proposta na Seção 4.2 foi usado um SoC FPGA da AMD-Xilinx, especificamente a placa ZYBO Z-20 (vide Figura 4.5), que comporta um chip XC7Z020-1CLG400C. Os recursos disponíveis da placa podem ser observadas na Tabela 4.1. Para as etapas do algoritmo executado em ARM, novamente foi utilizado um sistema *bare-metal*.



4.3.1 Supervisório de Alto Nível para Estímulo e Monitoramento do Respirador –SANER

Para o controle e teste do respirador, foram inicialmente desenvolvidos (pela equipe de pesquisa do respirador TICÊ) dois sistemas de interface humana:

- Tela *touch*: Tela para uso do profissional de saúde, que permite variar o modo de operação do respirador (PCV ou VCV) e permite selecionar os *setpoints* dos controladores de maneira a determinar a pressão/fluxo, concentração de oxigênio e PEEP. Também é possível escolher as frequências respiratórias do paciente.
- Supervisório de interface gráfica Phyton: Possui todas as funcionalidades da tela *touch* em uma interface gráfica organizada, além de permitir a seleção dos parâmetros dos controladores PID 0, 1 e 2 e o armazenamento de dados. O supervisório implementado em Phyton também permite a calibração do sensor de O₂ e o desligamento de dos controladores para uma operação manual do sistema.

Constatou-se, entretanto, a necessidade de desenvolver um programa ainda mais avançado de interface, pois, o supervisório em Phyton não permite a utilização de estímulos coordenados por *scripts*. Ou seja, a execução de ensaios com entradas senoidais ou com intervalos precisos de execução seria impossível. Além da manipulação livre das entradas do sistema, o sistema SANER também facilitou o processamento e a análise dos dados de saída. É importante ressaltar também que o sistema SANER (Desenvolvido em MATLAB) possui todas as funcionalidades apresentadas no supervisório Phyton.

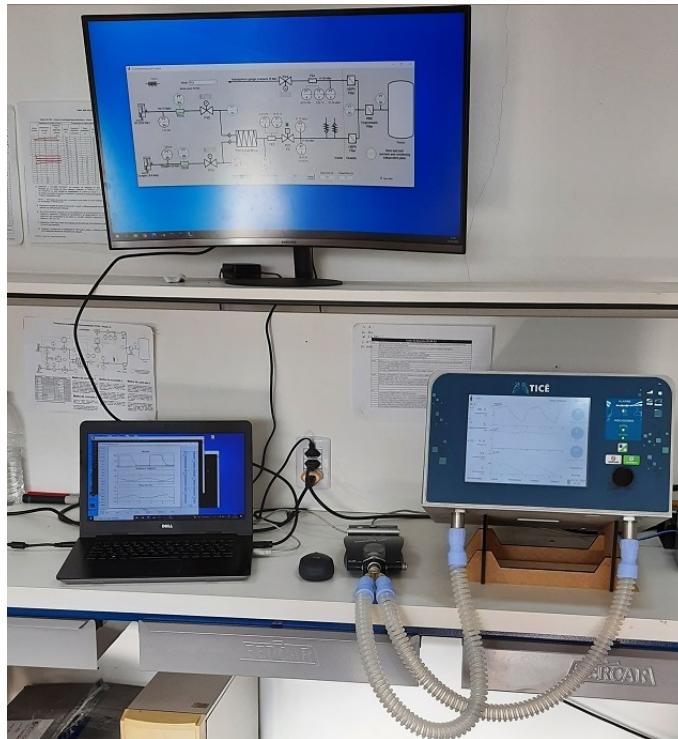


Figura 4.6 – Respirador mecânico TICE e interface de comunicação com computador.

4.3.2 Hiper-parâmetros NARX do modelo de pressão sintética

Para solucionar o problema de seleção dos hiper-parâmetros do modelo NARX, um código em MATLAB foi desenvolvido capaz de iterar sobre todas as combinações de na , nb e M dentro de um domínio pré-selecionado pelo usuário (Figura 4.2). Através do uso da métrica de correlação R2, a melhor combinação de hiper-parâmetros é automaticamente escolhida. Essa validação de dados é feita por método *One step ahead* e por *Free-run*, descritos na Seção 2.8. Para a execução desse código, também é necessária a seleção de alguns parâmetros manuais quem devem ser escolhidos pelo usuário (seja por experiência, por referência, ou por tentativa e erro). Esses parâmetros são os seguintes:

- $na_min=1$: Limite inferior do número de atrasos da saída;
- $nb_min=1$: Limite inferior do número de atrasos da entrada;
- $na_max=30$: Limite superior do número de atrasos da saída;
- $nb_max=30$: Limite superior do número de atrasos da entrada;
- $M=20$: Limite máximo do número de neurônios utilizados pela rede RBF.

Código 4.2 – Loop de iteração de hiper-parâmetros RBF

```

1 for i=1:M
2   for na=na_min:na_max
3     for nb=nb_min:nb_max
4       [R2_T]=RBF_funct(ue, ye, uv, yv, na, nb, M)

```

```

5      RBF_iter_results(i,1)=MAX_TERMOS;
6      RBF_iter_results(i,2)=na;
7      RBF_iter_results(i,3)=nb;
8      RBF_iter_results(i,4)=l;
9      RBF_iter_results(i,5)=norm_factor;
10     RBF_iter_results(i,6)=R2_T;
11
12     save('RBF_iter_results_1_4')
13 end

```

Uma vez selecionados os hiper-parâmetros manuais, é necessária a preparação dos dados de treinamento e validação, através de ensaios feitos em laboratório (Seção seguinte).

4.3.3 Ensaio para identificação do sistema de pressão sintética

Através do uso de diversos ensaios com estímulos multi-seno nas válvulas de entrada, foi possível levantar um *dataset* de entradas e saídas para uma identificação de sistema caixa-preta. Os ensaios foram executados da seguinte maneira:

1. Selecionar valores fixos de inspiração e expiração: 1s e 2s respectivamente.
2. Desligar todos os controladores do sistema.
3. Gerar ondas multiseno aleatórias com variabilidade de amplitude e frequência respeitando os limites das válvulas FV2, FV3 e FV4.
4. Iniciar o monitoramento contínuo de saídas.

Para cada ensaio laboratorial foi obtido um conjunto de dados com 8000 variáveis de entrada, utilizando um tempo de amostragem de 0.02s.

No total, trinta (30) ensaios diferentes foram realizados, com condições externas e pulmões artificiais de complacência marginalmente diferentes. Desses 30 ensaios, apenas um foi escolhido dando preferência ao ensaio com maior variabilidade de entradas e menor perda de dados na interface USB respirador-computador.

4.4 Considerações finais do capítulo

Neste capítulo, foram apresentadas as arquiteturas de *hardware* desenvolvidas para os dois primeiros estudos de caso escolhidos: estimativa de estresse em materiais e detecção de falhas de solda em tubulações. Também foram detalhados os algoritmos embarcados escolhidos para aceleração em *hardware*, assim como os dados utilizados para a execução desses algoritmos. Além disso, o capítulo descreveu as melhorias desenvolvidas para otimização da arquitetura, através de uma análise crítica dos resultados do primeiro estudo de caso.

Ainda neste capítulo, foi descrito o processo de coleta de dados do terceiro estudo de caso, através da bancada real do respirador mecânico, utilizado para a identificação dos

valores de pressão sintética. Também foi apresentado o método desenvolvido para estimação dos hiperparâmetros e parâmetros NARX e RBF do modelo do respirador.

5 Resultados

5.1 Primeiro e Segundo Estudo de Caso - Arquitetura HW/SW Co-design

5.1.1 Desempenho da Arquitetura Proposta para o Primeiro Estudo de Caso

5.1.1.1 Consumo de recursos

Inicialmente, a arquitetura com 100 *cores* foi projetada para tentar maximizar o paralelismo e a taxa de transferência, apresentando um compromisso entre desempenho e área ocupada no chip na implementação de *hardware*.

A Figura 5.1 apresenta a implementação do dispositivo, permitindo uma observação mais visual da quantidade de recursos utilizados, com destaque para as células dos componentes RTL propostos. A solução baseada em HDL com 100 *cores* utiliza quase toda a lógica disponível no dispositivo, como confirmado pelos dados da Tabela 5.1. Além da ocupação de recursos da arquitetura geral, a tabela discrimina o consumo de recursos dos módulos RTL desenvolvidos.

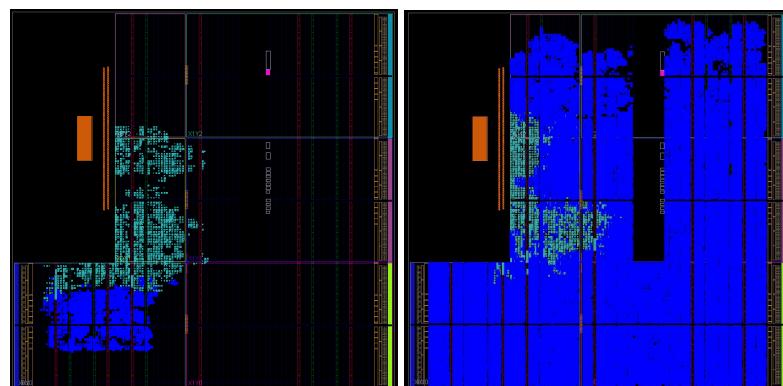


Figura 5.1 – Visão de implementação do dispositivo. Esquerda: HDL com 5 *cores*; Direita: HDL com 100 *cores*.

Tabela 5.1 – Utilização de *hardware* da arquitetura geral e dos módulos RTL implementados no chip xc7z020clg400-1.

Informações do Design	LUT (53200)	FF (106400)	BRAM (140)	DSP (220)
HDL com 5 núcleos	4794 (9,01%)	4817 (4,53%)	3,5 (2,5%)	10 (4,55%)
• Módulos RTL	1789 (3,36%)	1291 (1,21%)	0 (0%)	10 (4,55%)
HDL com 100 núcleos	44935 (84,46%)	26140 (24,57%)	3,5 (2,5%)	200 (90,91%)
• Módulos RTL	41930 (78,81%)	22615 (21,26%)	0 (0%)	200 (90,91%)

5.1.1.2 Velocidade de execução

A latência computacional, apresentada na Tabela 5.2, foi registrada utilizando um AXI-Timer durante a verificação em circuito implementado na placa FPGA. Observando a tabela, pode-se notar que a arquitetura ARM teve a maior latência (19,79 ms), sendo cerca de 11,7 vezes mais lenta do que a implementação HDL com 5 núcleos e 14 vezes mais lenta do que a com 100 núcleos. Outro ponto a ser considerado para essa arquitetura é que o aumento no número de núcleos da arquitetura HDL diminui a latência computacional, porém, o ganho entre 5 e 100 núcleos é pequeno (16,2% de melhoria). A Tabela 5.2 apresenta também a latência computacional para o conjunto completo de teste, consistindo de 250 execuções.

Tabela 5.2 – Latência computacional medida

Latência computacional	Arquitetura HDL (5 núcleos)	Arquitetura HDL (100 núcleos)	ARM
Execução única [ms]	1.688	1.414	19.792
Conjunto de testes completo [ms]	422.0	353.5	4948.0

Para avaliar a eficiência computacional, a Tabela 5.3 exibe o relatório de consumo de energia considerando o consumo dinâmico e estático de energia das partes PS (Processador de Sistema) e PL (Lógica Programável) das arquiteturas analisadas. A partir dos dados, observa-se que a arquitetura HDL com 100 *cores* consome quase 33.39% mais energia do que a arquitetura com 5 *cores*,

Para contextualizar os dados, a eficiência de cada arquitetura proposta foi avaliada em termos de frequência de inferência (f_L) por watt. A arquitetura HDL com 100 *cores* atinge cerca de 336,6 f_L/W , enquanto a arquitetura HDL com 5 *cores* alcança aproximadamente 376,1 f_L/W , sendo, portanto, 12% mais eficiente.

Tabela 5.3 – Resumo do relatório de potência.

Implementação on-chip	HDL 5 núcleos	HDL 100 núcleos
Dinâmico (W)	1.436	1.948
• PS Dinâmico (W)	1.404	1.404
• PL Dinâmico (W)	0.032	0.544
◦ Projeto Proposto (W)	0.006	0.521
Estático do Dispositivo (W)	0.138	0.153
Potência Total no Chip (W)	1.575	2.101

5.1.2 Desempenho da Arquitetura Proposta para o Segundo Estudo de Caso

5.1.2.1 Consumo de recursos

Para a arquitetura de co-design HW/SW, apenas o cálculo da matriz reduzida (através do PCA) foi acelerado pelo PL, conforme indicado na Seção 4.2.1. O restante do algoritmo é executado em *software* (ARM), ou seja, em PS. A implementação da arquitetura de HW/SW co-design foi aplicada de acordo com a Seção 4.2.

A Tabela 5.3 apresenta o gasto de energia do PS e do PL na arquitetura geral HDL, embora a arquitetura RTL tenha sido responsável por 98.5% do processamento necessário (como discutido na Seção 4.2.1), ela utilizou apenas 9.25% do consumo total de energia do dispositivo.

Tabela 5.4 – Resumo do relatório de consumo de energia

Componente	HDL 5 núcleos
Dinâmico (W)	1.438
• Dinâmico do PS (W)	1.404
• Dinâmico do PL (W)	0.034
◦ Design Proposto (W)	0.007
Estático do Dispositivo (W)	0.139
Potência Total no Chip (W)	1.577

O relatório de utilização de recursos apresentado na Tabela 5.5 confirma o baixo uso dos recursos disponíveis na placa ZYBO-Z20. Também é possível de se observar que apenas uma pequena parcela do gasto de energia dinâmico PL é feita pelo design proposto HDL, o restante do consumo de energia PL é feito pelos módulos de interconexão AXI e pelo módulo AXI-DMA.

Tabela 5.5 – Utilização de *hardware* do design implementado

Informações do Design	LUT (53200)	FF (106400)	BRAM (140)	DSP (220)
HDL com cinco núcleos	5158 (9.7%)	4598 (4.32%)	3.5 (2.5%)	10 (4.54%)
• Módulos RTL	2136 (4.01%)	1007 (0.94%)	0 (0%)	10 (4.54%)

5.1.2.2 Velocidade de execução

A Tabela 5.6 apresenta os tempos de execução do algoritmo implementado em 4 plataformas diferentes. A tabela foi dividida em 3 colunas, representando as diferentes etapas de execução do algoritmo: A primeira coluna apresenta os tempos de carregamento de todas as imagens na memória RAM dos respectivos dispositivos. A segunda coluna

apresenta o tempo de execução para o processamento de todas as imagens. Já a terceira coluna, apresenta o tempo de execução para processar uma imagem do conjunto de validação. Para essa análise de performance, o tempo de carregamento das imagens não foi analisado para a arquitetura HDL proposta (N/A), pois a análise foi feita a partir do momento em que os dados necessários já estavam alocados de maneira ideal na memória DDR3 do processador ARM.

Vale ressaltar que os dispositivos ARM e HDL foram analisados em *bare-metal*, executando o algoritmo proposto de maneira exclusiva. Já o processador Intel i5, executou o algoritmo através de um sistema operacional Windows 10 de 64 bits.

Tabela 5.6 – Resultados experimentais da implementação de *hardware*.

	Tempo de carregamento (s)	Tempo para processar todas as imagens (s)	Tempo para processar uma imagem (s)
Intel i5 (1035G7) com SSD	45.475	110.74	0.0184
ARM 1 núcleo	418.69	2277.09	0.3793
ARM 2 núcleos	418.69	1225.25	0.2041
Arquitetura HDL proposta	N/A	1.9679	0.00033

Vale ressaltar que os resultados numéricos foram iguais independente da plataforma utilizada. Ou seja, as imagens que foram classificadas como '*tubo danificado*' foram as mesmas, não apresentando resultados diferentes para o algoritmo executado puramente em SW e para o algoritmo executado na arquitetura proposta. Esse é o resultado esperado, visto que todas as etapas da arquitetura proposta e do código em C utilizaram a mesma representação numérica de ponto flutuante de 32 bits.

5.2 Terceiro Estudo de Caso - Respirador Mecânico

5.2.1 Dados disponíveis do sistema respirador mecânico

O respirador mecânico TICE possui uma interface de comunicação USB serial com *Baud Rate* de 256000 bits/s, que permite ao usuário controlar as válvulas do sistema pneumático, modificar o comportamento dos controladores/preditores e coletar dados de diversos sensores (Figura 4.6). Algumas das variáveis disponíveis para leitura via USB são:

- *Fv1*: % de abertura da válvula de entrada de AR do paciente
- *Fv2*: % de abertura da válvula de ar comprimido
- *Fv3*: % de abertura da válvula de O2 concentrado

- $Fv4$: % de abertura da válvula de saída de AR do paciente
- $Ft4$: Fluxo na saída de ar do paciente
- $At3$: Concentração de O₂ em % na linha de entrada de AR do paciente (não compensado)
- $PT2$: Pressão na linha de entrada de AR do paciente

A Figura 5.2 apresenta os valores coletados das variáveis de interesse.

5.2.2 Identificação do sistema de pressão sintética

Através da execução do ensaio de identificação proposto na Subseção 4.3.3, foram montadas as matrizes necessárias para o algoritmo de identificação RBF:

1. $ue = 7000$ primeiras amostras das variáveis de entrada (treinamento)
2. $ye = 7000$ primeiras amostras da variável de saída $PT2$ (treinamento)
3. $uv = 1000$ ultimas amostras das variáveis de entrada (validação)
4. $yv = 1000$ últimas amostras da variável de saída $PT2$ (validação)

Após a execução do algoritmo descrito na Seção 4.2, foi encontrada a melhor combinação dos hiper-parâmetros na , nb , M e parâmetros c e W (centroides e peso da rede RBF, respectivamente).

Os melhores valores de coeficiente R2 foram 0.817 para validação em OSA e 0.6659 para validação em FR utilizando os hiper-parâmetros $na = 2$, $nb = 3$, e $M = 13$. Os valores de estimativa da pressão PT2 e valores dos coeficientes de correlação ao longo das amostras para estimações OSA e FR podem ser observados nas Figuras 5.3, 5.4, 5.5, e 5.6.

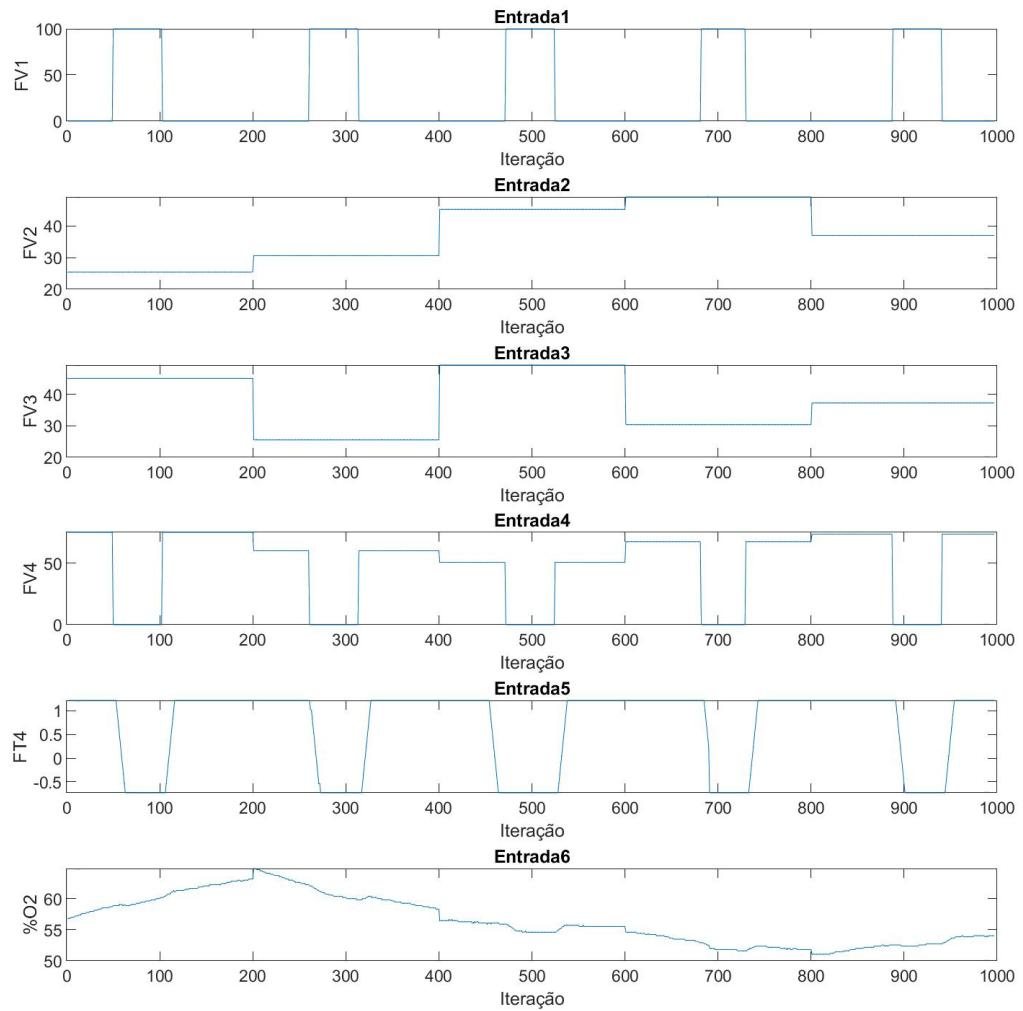


Figura 5.2 – Entradas utilizadas para validação (FR e OSA).

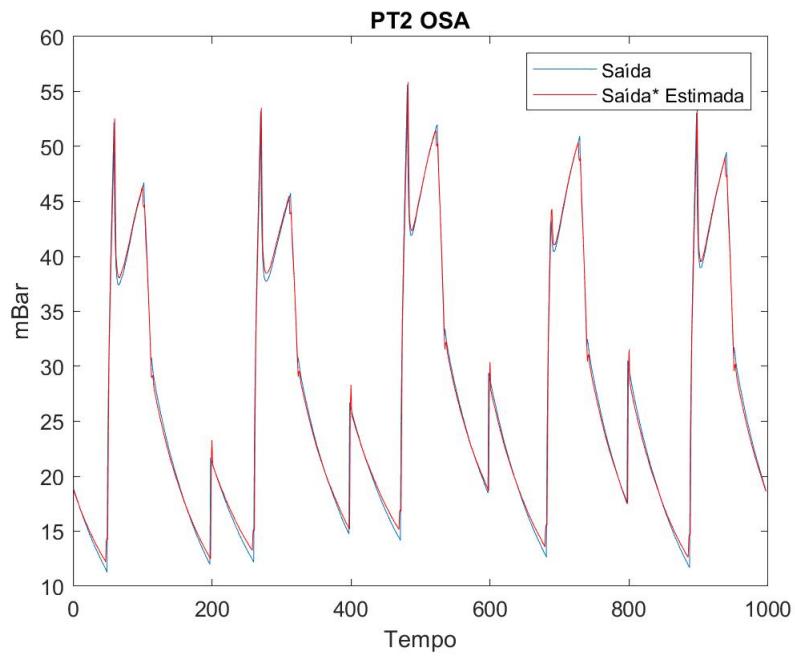


Figura 5.3 – Estimação da pressão PT2 OSA.

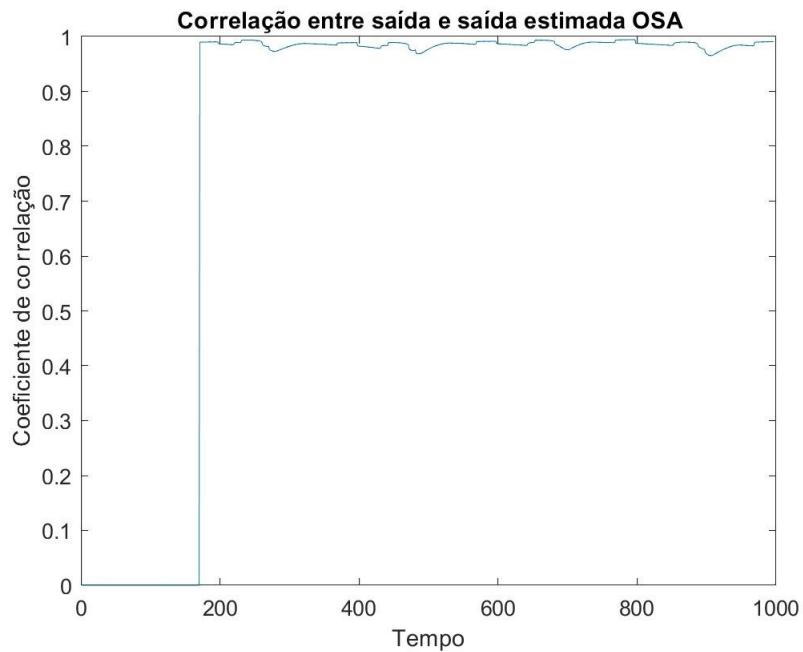


Figura 5.4 – Correlação móvel OSA (janela de 200 dados).

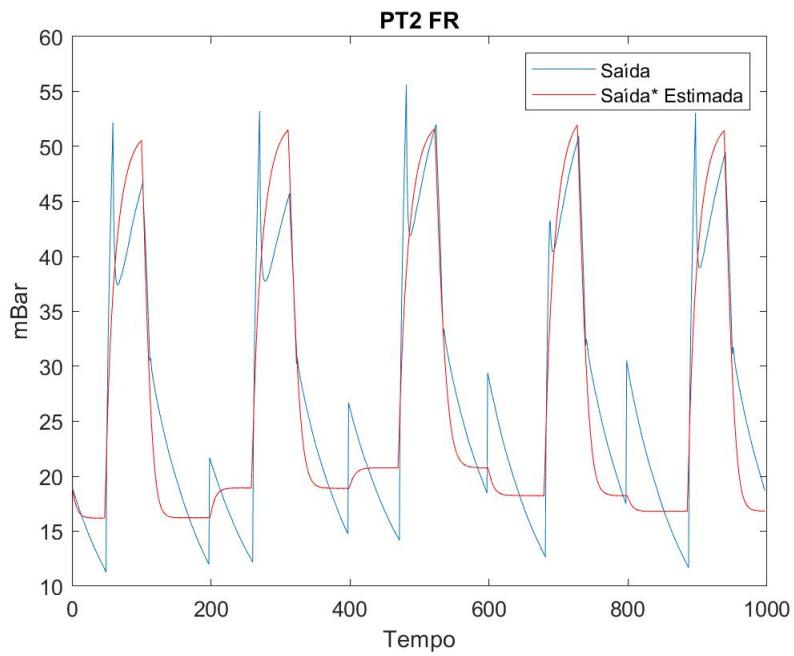


Figura 5.5 – Estimação da pressão PT2 FR.

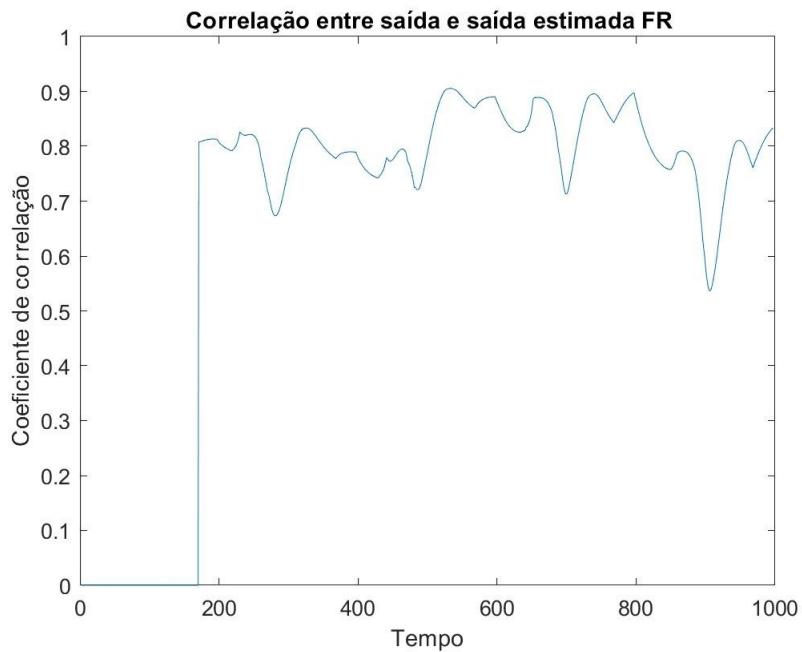


Figura 5.6 – Correlação móvel FR (janela de 200 dados).

O modelo, apesar de compacto (somente 3 *delays* de saída e 2 *delays* de entrada, vide Figura 5.7), apresentou boa capacidade de estimação de saída em *free-run* ($R^2 = 0.66$).

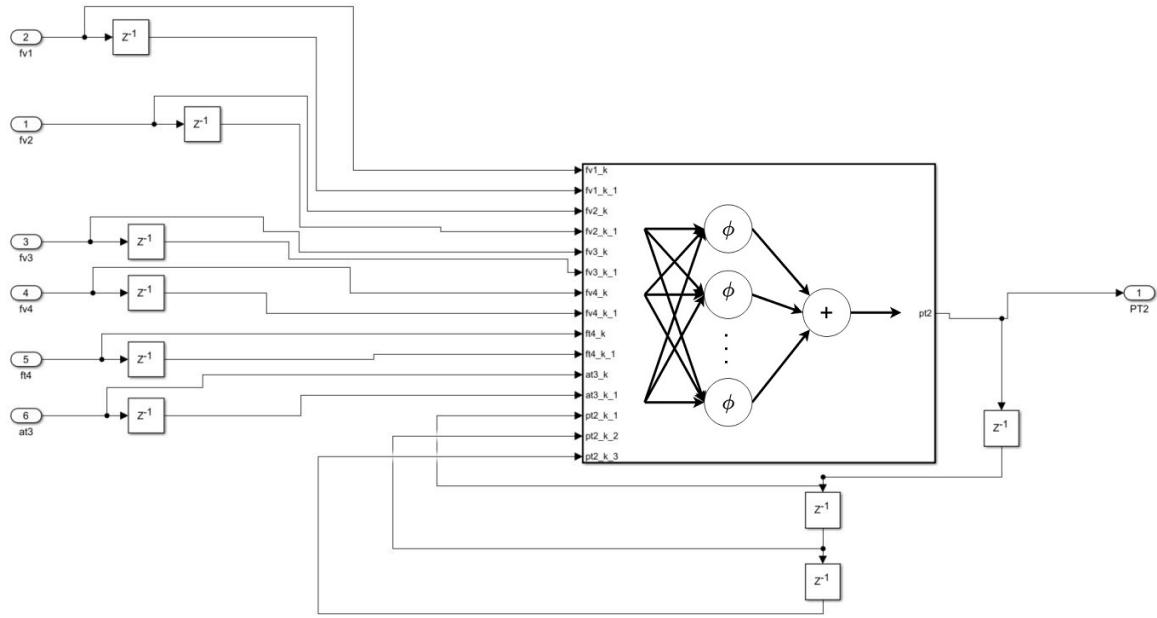


Figura 5.7 – Modelo no simulink com hiper-parâmetros NARX definidos ($na=3$, $nb=2$).

5.2.3 Proposta de Implementação Embarcada do Terceiro Estudo de Caso

Como opção de aceleração de processamento do modelo apresentado no terceiro estudo de caso, pode-se utilizar uma implementação em *hardware* da rede neural RBF utilizada (Figura 5.8). Para isso, existem soluções consolidadas na literatura que apresentam ganhos significativos de tempo de processamento (Tonussi; Quintero, 2020; Sampaio *et al.*, 2016).

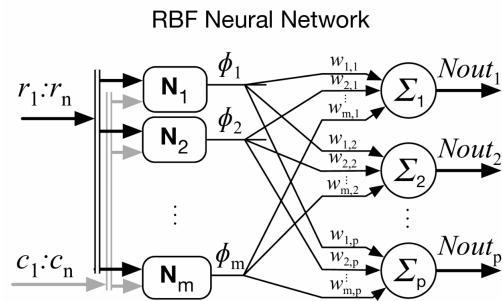


Figura 5.8 – Esquemático do design da rede neural RBF elaborada por (Sampaio *et al.*, 2016).

Por exemplo, no trabalho (Tonussi; Quintero, 2020), é apresentada uma redução do tempo de processamento de cerca de 228% em relação a implementação do mesmo algoritmo em código C em ARM. Apesar dessas soluções apresentarem ganhos significativos, por conta do respirador mecânico não possuir um módulo FPGA integrado ao hardware, a solução de aceleração em *hardware* da rede neural RBF é incompatível com o projeto atual do respirador mecânico TICÊ apresentado na Seção 2.9.

5.3 Discussões sobre os Resultados

5.3.1 Resultados do Primeiro Estudo de Caso

Observando os resultados apresentados pela arquitetura desenvolvida para o primeiro estudo de caso, pode-se observar pela Figura 5.1 e pela Tabela 5.1, que o consumo de recursos na placa foi consideravelmente baixo, para implementação com 5 *cores*, utilizando cerca de 5% do total de recursos disponíveis. Já para implementação com 100 cores, o consumo de recursos foi quase completo, utilizando cerca de 90% dos DSPs e LUTs. Além do alto consumo de recursos, a implementação em 100 *cores* também apresentou consumo de energia consideravelmente maior (33%). Apesar do consumo de recursos ter sido muito diferente entre a arquitetura de 5 e 100 *cores*, a redução do tempo de processamento entre as implementações não apresentou um ganho linear (apenas 16,2% mais rápido).

Projetando uma possível implementação da arquitetura em um dispositivo embarcado comercial, estima-se que a arquitetura de 5 *cores* apresentou uma relação mais vantajosa entre custo computacional e velocidade de execução.

Dito isso, apesar do tempo de processamento entre as arquiteturas de 5 *cores* e 100 *cores* ter sido similar, o tempo de processamento do mesmo algoritmo em ARM foi cerca de 12 vezes mais lento quando utilizada para execução da etapa de inferência do algoritmo de monitoramento de estresse em estruturas.

Finalizada a execução do algoritmo de identificação de falhas, a precisão apresentada foi de 100% de acerto da classificação binária de falhas de um conjunto de 6003 leituras diferentes, que compõem os dados de validação. O mesmo resultado também foi alcançado no trabalho original de (Holguin; Ayala; Kubrusly, 2022).

5.3.2 Resultados do Segundo Estudo de Caso

Uma vez analisados os resultados, uma análise crítica foi feita sobre possíveis melhorias na arquitetura apresentada no estudo de caso anterior. Através do estudo de métodos de aceleração de arquiteturas em FPGA, a arquitetura foi melhorada e aplicada para um estudo de caso diferente (detecção de falhas em soldas). Para esse segundo estudo de caso, além da arquitetura implementada e do processador ARM, também foi feito um estudo de performance comparativo com um processador comercial Intel, e com um processador ARM *dual-core*.

Ao inspecionar a Tabela 5.6, pode-se observar o quanto mais rápida foi a arquitetura HDL proposta em comparação com a implementação baseada em ARM. Apesar da FPGA operar a apenas 15% do clock do ARM, ele alcançou o resultado 625 vezes mais rápido. O ganho da arquitetura foi evidente, mesmo quando comparado com dispositivos capazes de paralelização de tarefas em *software* através de processadores multi-core. O tempo de execução da etapa de inferência do algoritmo de detecção de falhas em soldas foi executado

em apenas 2 segundos na arquitetura proposta, viabilizando um possível monitoramento *online* da tubulação.

Como limitação da arquitetura proposta, é importante informar que o desempenho no design HDL somente foi possível de ser atingido quando todos os dados necessários para a execução do PCA estavam devidamente alocados e sequenciados na memória do sistema (PS), permitindo o tempo ideal de transferência de um ciclo de clock pela interface AXI-Stream.

Assim, como no primeiro estudo de caso, os resultados de identificação de quais imagens apresentaram falhas em soldas foram os mesmos, independentemente da plataforma utilizada, que classificou as imagens com 100% de precisão. Resultado similar ao trabalho apresentado em ([Virkkunen et al., 2021](#)), porém, executado em uma fração do tempo de um processador convencional.

5.3.3 Resultados do Terceiro Estudo de Caso

Através da execução do código descrito na Subseção [4.3.2](#), foram obtidos os melhores hiper-parâmetros do modelo NARX e RBF utilizados na identificação do respirador. Para avaliar a qualidade dos hiper-parâmetros de cada iteração, um conjunto de entradas de validação foi utilizado (Figura [5.2](#)), em conjunto com o critério estatístico R^2 , aplicado sobre a fidelidade da saída estimada de pressão na linha inspiratória. Para o melhor conjunto de hiper-parâmetros, o valor de R^2 encontrado foi de 0,82 em validação por OSA e 0,66 para validação em *free-run*. Este é um resultado satisfatório, pois o modelo a ser identificado possui diversas não linearidades que não foram devidamente modeladas. Algumas dessas não linearidades são: Complacência do pulmão do paciente, resistências na linha pneumática e imprecisões dos sensores.

Uma das limitações importantes sobre a estimativa de pressão sintética, foi que o uso do modelo identificado ainda é inviável em um sistema embarcado, visto que é necessário um alto custo computacional para estimativas de pressão em tempo real. Neste trabalho, todas as estimativas do modelo do respirador foram feitas em *SIMULINK*, de maneira *offline*. Conclui-se, portanto, que através da utilização do modelo identificado em uma estrutura similar à apresentada na Figura [2.5](#), é possível de monitorar possíveis falhas de sensoriamento de pressão inspiratória do paciente.

6 Conclusões

6.1 Primeiro e Segundo Estudo de Caso

Neste trabalho, foram apresentadas soluções embarcáveis para identificação de falhas em três estudos de caso diferentes. Na primeira etapa de desenvolvimento deste trabalho, foi feita a implementação do algoritmo PCA puramente em C, utilizado para os dois primeiros estudos de caso. Esse algoritmo foi embarcado de maneira *stand-alone* em um ARM no SoC FPGA Zynq 7020 da placa ZYBO-Z20, operando em modo *single-core* e *dual-core*, apresentando um ganho de performance de quase 100% entre os modos de operação do ARM. Essa implementação puramente em *software*, também permitiu um *profiling* do algoritmo, de maneira que as seções de maior custo computacional foram possíveis de serem selecionadas para implementação em *hardware*. Após a constatação de que o maior custo computacional do algoritmo PCA era um procedimento de multiplicação de matrizes, a **primeira hipótese** pode ser confirmada, visto que multiplicação matricial é uma tarefa com grande oportunidade de paralelização em *hardware*.

Uma vez determinado quais etapas do algoritmo devem ser implementadas em *hardware*, uma arquitetura inédita foi desenvolvida, utilizando uma máquina de estados finitos sequencial e estratégias de *hardware/software co-design*. Também foram avaliadas a performance da arquitetura utilizando 5 e 100 multiplicadores paralelos, que apresentaram um ganho de 1170% e 1400% respectivamente quando comparados com o tempo de execução da arquitetura em ARM. Também foi avaliada a eficiência energética da arquitetura, apresentando uma frequência de inferência por Watt de 376.1 e 336.6 f_L/W para as arquiteturas com 5 e 100 multiplicadores paralelos, respectivamente. Esses valores foram significativamente superiores à eficiência energética do ARM, que apresenta apenas 0.13 f_L/W .

Para o segundo estudo de caso apresentado neste trabalho, foram implementadas melhorias na arquitetura utilizada para a aceleração do processamento do PCA. Dentre essas melhorias, a mais significativa foi a reestruturação da arquitetura para um modelo de *pipeline*, aumentando o *throughput* de ~0.5 Milhões de estimativas por segundo para ~50 milhões de estimativas por segundo. Como resultado, a nova arquitetura apresentou um ganho de desempenho de 625 vezes em relação a implementação puramente em *software* (ARM embarcado). Apesar do ganho significativo de desempenho, a arquitetura desenvolvida resultou em um aumento de apenas 3% no consumo total de energia, confirmando de maneira objetiva a **segunda hipótese** do trabalho.

Visto que o período de inferência do algoritmo de detecção de falhas em soldas foi reduzido de 2277 segundos (ARM *single-core*) para 2 segundos, é possível concluir que a arquitetura proposta viabilizou a execução do algoritmo em tempo real de maneira embarcada,

validando a **terceira hipótese**.

Apesar dos objetivos terem sido alcançados no estudo de caso de monitoramento de dutos, a solução proposta é voltada exclusivamente para a detecção de danos, sem estimar sua gravidade (profundidade da trinca). Isto é uma limitação deste trabalho, visto que fornecer informações sobre a gravidade da trinca pode ser essencial para monitorar seu crescimento e planejar a manutenção.

Sobre a qualidade dos resultados, o primeiro e o segundo estudo de caso apresentaram resultados similares de identificação de falhas independente da plataforma utilizada (ARM, Intel ou Arquitetura *HW/SW*), comprovando que a integridade dos dados foi mantida na arquitetura desenvolvida.

6.2 Terceiro Estudo de Caso

No terceiro estudo de caso foi desenvolvido um modelo para identificar falhas em um sensor de pressão na linha inspiratória de um respirador mecânico. Para isso, foi utilizado um método de identificação de falhas através da comparação entre o valor estimado do sensor de pressão do ventilador e o valor real medido. O modelo do sensor foi desenvolvido de maneira não determinística, utilizando de uma rede neural RBF alimentada com entradas de parâmetros NARX. Através do uso de algoritmos que iteram entre os hiper-parâmetros de uma rede neural RBF, um modelo final foi proposto, o qual apresentou uma correlação de 0.7 com os valores reais medidos em validação *Free-Run* por critério estatístico R2. Dessa forma, foi possível uma estimação útil dos valores de pressão sintética, validando a **quarta hipótese**. Infelizmente, apesar do modelo ser suficientemente representativo para a finalidade proposta, o custo computacional da estimação da pressão sintética em tempo real inviabilizou seu uso em plataforma embarcada no produto respirador mecânico TICÊ. Apesar disso, uma proposta de arquitetura embarcada baseada em co-projeto *hardware/software* é apresentada como possível alternativa e trabalho futuro.

6.3 Trabalhos Futuros

A seguir se relacionam propostas de trabalhos futuros.

- Utilização da arquitetura *HW/SW* proposta em outros problemas de redução de dimensionalidade matricial. Isso se justifica pelo fato de que técnicas de processamento de imagens por ultrassom são amplamente aplicáveis, indo além do monitoramento de estruturas e materiais, como evidenciado no estudo bibliográfico apresentado no capítulo 3.1.
- Desenvolvimento de uma plataforma completa para a implementação da arquitetura proposta, incluindo funcionalidades de transferência de dados. Isso permitiria a

realização de testes em monitoramento de dutos reais, utilizando a arquitetura de maneira *online*.

- Uma área de aprimoramento do terceiro estudo de caso seria a aplicação de técnicas de aceleração de processamento da rede neural RBF em *hardware*, assim como apresentado na Seção 5.2.3 embarcando a arquitetura proposta no respirador mecânico, permitindo a verificação em tempo real de integridade do sinal de pressão real.

Referências

- ABBAS, S.; LI, F.; QIU, J.; ZHU, Y.; TU, X. Optimization of ultrasonic guided wave inspection in structural health monitoring based on thermal sensitivity evaluation. **Journal of Low Frequency Noise Vibration and Active Control**, v. 40, n. 1, p. 601 – 622, 2021. Cited by: 8; All Open Access, Gold Open Access. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85077636739&doi=10.1177%2f1461348419886189&partnerID=40&md5=de6aabaf37aac6b9e2d198d98bba50d9>. Citado na p. 40.
- AHMED, K.; SHAH, A. A.; WANG, L.; WANG, S. Modeling and identification of power forecasting scheme for real pv system using grey box neural network based narmax model. In: **2022 19th International Bhurban Conference on Applied Sciences and Technology (IBCAST)**. [S.l.: s.n.], 2022. p. 562–567. Citado na p. 31.
- ASHWATH; SUNDARESAN, J.; GEOFFREY; CHASE, G.; M; SHAW, Y.; SHIONG; CHIEW, T.; DESAIVE. Model-based optimal peep in mechanically ventilated ards patients in the intensive care unit. **BioMedical Engineering OnLine**, v. 10, n. 64, 2011. Citado nas pp. 44 e 46.
- ASTM International. **Standard Practice for Conducting Force Controlled Constant Amplitude Axial Fatigue Tests of Metallic Materials**. West Conshohocken, PA: ASTM International, 2015. Disponível em: <https://www.astm.org/e0466-15.html>. Citado na p. 23.
- ASTM International. **Standard Test Methods for Tension Testing of Metallic Materials**. West Conshohocken, PA: ASTM International, 2016. Disponível em: https://www.astm.org/e0008_e0008m-16a.html. Citado na p. 23.
- ASTM International. **Standard Test Method for Measurement of Fracture Toughness**. West Conshohocken, PA: ASTM International, 2020. Disponível em: <https://www.astm.org/e1820-20.html>. Citado na p. 23.
- AZIZI, T. Comparative analysis of statistical, time-frequency, and svm techniques for change detection in nonlinear biomedical signals. **Signals**, v. 5, n. 4, p. 736–755, 2024. ISSN 2624-6120. Disponível em: <https://www.mdpi.com/2624-6120/5/4/41>. Citado na p. 18.
- BARBALHO, S. C. M. **Desenvolvimento de respirador mecânico de baixo custo com sistemas de controle de volume e pressão e adequado às condições sanitárias para pacientes em UTI devido ao COVID-19**. 2021. Projeto FAPDF - UnB. Citado nas pp. 10 e 34.
- BARILE, G.; LEONI, A.; PANTOLI, L.; STORNELLI, V. Real-time autonomous system for structural and environmental monitoring of dynamic events. **Electronics**, v. 7, n. 12,

2018. ISSN 2079-9292. Disponível em: <https://www.mdpi.com/2079-9292/7/12/420>. Citado na p. 16.
- BEAULIEU, A.; BOSSE, D.; MICHEAU, P.; AVOINE, O.; PRAUD, J.-P.; WALTI, H. Measurement of fractional order model parameters of respiratory mechanical impedance in total liquid ventilation. **IEEE Transactions on Biomedical Engineering**, v. 59, n. 2, p. 323–331, 2012. Citado nas pp. 44, 46 e 48.
- BENGIO, Y.; COURVILLE, A.; VINCENT, P. Representation learning: A review and new perspectives. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, Institute of Electrical and Electronics Engineers (IEEE), 2013. Citado na p. 28.
- BENTHOWAVE. **BII Phased Array Ultrasonic System**. 2025. Accessed: 2025-03-05. Disponível em: <https://www.benthowave.com/products/BII-7630PhasedArray.html>. Citado nas pp. 10 e 24.
- BESTARD, G. A. **Sensor fusion and embedded devices to estimate and control the depth and width of the weld bead in real time**. Tese (Doutorado) — Universidade de Brasília, 2017. Citado nas pp. 10, 35 e 36.
- BILLINGS, S. A.; VOON, W. S. Nonlinear system identification: Narmax modeling. **International Journal of Control**, v. 40, n. 2, p. 203–220, 1984. Citado nas pp. 31 e 33.
- BORRELLO, M.; ENTERPRISES, T.; DIEGO, S. Modeling and control of systems for critical care ventilation. IEEE, p. 2166–2180, 2005. Citado na p. 47.
- BOUZENAD, A. E.; MOUNTASSIR, M. E.; YAACOUBI, S.; DAHMENE, F.; KOABAZ, M.; BUCHHEIT, L.; KE, W. A semi-supervised based k-means algorithm for optimal guided waves structural health monitoring: A case study. **Inventions**, v. 4, n. 1, 2019. Cited by: 37; All Open Access, Gold Open Access. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85068050554&doi=10.3390%2finventions4010017&partnerID=40&md5=1a3853ba688160cc9124bec19456fd1e>. Citado na p. 42.
- BRATH, A. J.; SIMONETTI, F.; NAGY, P. B.; INSTANES, G. Guided wave tomography of pipe bends. **IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control**, v. 64, n. 5, p. 847 – 858, 2017. Cited by: 26; All Open Access, Green Open Access. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85020050102&doi=10.1109%2fTUFFC.2017.2683259&partnerID=40&md5=69cc2c92f7373ff2c675d2817f251d63>. Citado na p. 42.
- BRUNTON, S. L.; KUTZ, J. N. **Data-Driven Science and Engineering**. [S.l.]: Cambridge University Press, 2019. Citado na p. 28.
- CHATBURN, R. L.; MIRELES-CABODEVILA, E. Closed-loop control of mechanical ventilation: Description and classification of targeting schemes. **Respiratory Care**, v. 56, 2011. ISSN 00201324. Citado na p. 46.

- CHEN, Z.; CHEN, Y.; HUANG, Q. Development of a wireless and near real-time 3d ultrasound strain imaging system. **IEEE Transactions on Biomedical Circuits and Systems**, v. 10, n. 2, p. 394 – 403, 2016. Cited by: 43. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84941922699&doi=10.1109%2fTBCAS.2015.2420117&partnerID=40&md5=ac4586e70976a5440c53bba4b267791e>. Citado nas pp. 39 e 40.
- DEHGHAN-NIRI, E.; SALAMONE, S. A multi-helical ultrasonic imaging approach for the structural health monitoring of cylindrical structures. **Structural Health Monitoring**, v. 14, n. 1, p. 73 – 85, 2015. Cited by: 57. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84919629937&doi=10.1177%2f1475921714548937&partnerID=40&md5=07b8902147787bba60d550c77b359765>. Citado na p. 42.
- DIGILENT. **Foto Zybo.** 2017. <https://store.digilentinc.com/zybo-z7-zynq-7000-arm-fpga-soc-development-board/>. Online; Acesso: 2020-01-12. Citado nas pp. 10 e 55.
- DUBUC, B.; EBRAHIMKHANLOU, A.; SALAMONE, S. Localization of multiple acoustic emission events occurring closely in time in thin-walled pipes using sparse reconstruction. **Journal of Intelligent Material Systems and Structures**, v. 29, n. 11, p. 2362 – 2373, 2018. Cited by: 9. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85047390654&doi=10.1177%2f1045389X18770857&partnerID=40&md5=d2f40d2ecaa661daa8278c0141169789>. Citado na p. 42.
- FAN, Z.; BAI, K.; CHEN, C. Ultrasonic testing in the field of engineering joining. **The International Journal of Advanced Manufacturing Technology**, Springer, p. 1–26, 2024. Citado na p. 17.
- FERREIRA, G. R. B. Master's thesis, **DATA-DRIVEN ULTRASONIC NON-DESTRUCTIVE EVALUATION OF PIPES AND WELDS IN THE CONTEXT OF THE OIL AND GAS INDUSTRY**. Rio de Janeiro, RJ: [s.n.], 2022. Available at <https://doi.org/10.17771/PUCRio.acad.57224>. Citado na p. 17.
- FERREIRA, G. R. B.; TONUSSI, G. R. V.; NOZ, D. M. M.; LLANOS, C. H.; KUBRUSLY, A. C.; AYALA, H. V. H. System-on-chip machine learning pipeline crack detection method with ultrasonic data. **IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS**, 2025. Citado nas pp. 10, 17, 19, 21, 26, 28 e 29.
- FERREIRA, P. M.; MACHADO, M. A.; CARVALHO, M. S.; VIDAL, C. Embedded sensors for structural health monitoring: Methodologies and applications review. **Sensors**, v. 22, n. 21, 2022. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/22/21/8320>. Citado na p. 17.
- FORAKER, R. E.; YU, S. C.; GUPTA, A.; MICHELSON, A. P.; SOTO, J. A. P.; COLVIN, R.; LOH, F.; KOLLEF, M. H.; MADDOX, T.; EVANOFF, B.; DROR, H.; ZAMSTEIN, N.;

- LAI, A. M.; PAYNE, P. R. O. Spot the difference: comparing results of analyses from real patient data and synthetic derivatives. **JAMIA Open**, v. 3, n. 4, p. 557–566, 12 2020. ISSN 2574-2531. Disponível em: <https://doi.org/10.1093/jamiaopen/ooaa060>. Citado na p. 18.
- FROST, J. **Regression Analysis: An Intuitive Guide**. 1. ed. [S.l.: s.n.], 2019. Citado na p. 32.
- G1 Globo. **Respirador desenvolvido na UnB será comercializado para unidades de saúde**. 2022. <https://g1.globo.com/df/distrito-federal/df2/video/respirador-desenvolvido-na-unb-sera-comercializado-para-unidades-de-saude-12060016.ghtml>. Acessado em: 1 dez. 2024. Citado na p. 21.
- GAO, G.; DANG, R.; NOURI, A.; JIA, H.; LI, L.; FENG, X.; DANG, B. Sand rate model and data processing method for non-intrusive ultrasonic sand monitoring in flow pipeline. **Journal of Petroleum Science and Engineering**, v. 134, p. 30 – 39, 2015. Cited by: 37; All Open Access, Bronze Open Access. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84938053159&doi=10.1016%2fj.petrol.2015.07.001&partnerID=40&md5=07f3c87d335b3c5063ac00f3ae609a61>. Citado na p. 42.
- GUO, J.; FENG, S.; GAO, G.; WANG, R.; WANG, M.; TAO, C. Parameter identification of respiratory mechanics based on convolutional neural networks. In: **2023 2nd International Conference on Automation, Robotics and Computer Engineering (ICARCE)**. [S.l.: s.n.], 2023. p. 1–5. Citado na p. 18.
- HELLIER, C. **Handbook of Nondestructive Evaluation**. [S.l.]: McGraw-Hill Education, 2012. Citado na p. 23.
- HERNANDEZ, F. U.; MORGAN, S. P.; HAYES-GILL, B. R.; HARVEY, D.; KINNEAR, W.; NORRIS, A.; EVANS, D.; HARDMAN, J. G.; KORPOSH, S. Characterization and use of a fiber optic sensor based on pah/sio₂ film for humidity sensing in ventilator care equipment. **IEEE Transactions on Biomedical Engineering**, v. 63, n. 9, p. 1985–1992, 2016. Citado nas pp. 44 e 46.
- HOLGUIN, C. D. V.; AYALA, H. V. H.; KUBRUSLY, A. C. Improved stress estimation with machine learning and ultrasonic guided waves. **Experimental Mechanics**, Springer Science and Business Media LLC, 2021. Citado na p. 18.
- HOLGUIN, C. D. V.; AYALA, H. V. H.; KUBRUSLY, A. C. Improved stress estimation with machine learning and ultrasonic guided waves. **Experimental Mechanics**, v. 62, n. 2, p. 237–251, 2022. ISSN 1741-2765. Citado na p. 69.
- ID, W. P. K.; AMOS, J.; AZER, M.; BAKER, D.; BASHIR, R.; BEST, C.; BETHKE, E.; ID, S. A. B.; BRALTS, E.; ID, M. C.; DIETKUS, R.; DURACK, G.; ELBEL, S.; ELLIOTT, G.; FAVA, J.; GOLDENFELD, N.; GOLDSTEIN, M. H.; HAYES, C.; HERNDON, N.; JAMISON, S.; JOHNSON, B.; JOHNSON, H.; JOHNSON, M.; KOLACZYNSKI, J.; LEE, T.; MASLOV, S.; ID, D. J. M.; MILNER, D.; MOLLER, R.; MOSLEY, J.; MUSSER,

- A.; NEWBERGER, M.; NULL, D.; BRYAN, L. O.; OELZE, M.; LEARY, J. O.; PAGANO, A.; PHILPOTT, M.; PIANFETTI, B.; PILLE, A.; ID, L. P.; RICCONI, B.; RUBESSA, M.; RYLOWICZ, S.; SHIPLEY, C.; ID, A. C. S.; STEWART, B.; SWITZKY, R.; TAWFICK, S.; ID, M. W.; WHITE, K.; WIDLOSKI, E. M.; ID, E. W.; WOOD, C.; ID, A. R. W. Emergency ventilator for covid-19. p. 1–19, 2020. Disponível em: <http://dx.doi.org/10.1371/journal.pone.0244963>. Citado nas pp. 47 e 48.
- IRMAK, H.; ALACHIOTIS, N.; ZIENER, D. An energy-efficient fpga-based convolutional neural network implementation. In: **2021 29th Signal Processing and Communications Applications Conference (SIU)**. [S.l.: s.n.], 2021. p. 1–4. Citado na p. 19.
- JI, Q.; JIAN-BIN, L.; FAN-RUI, L.; JIAN-TING, Z.; XU, W. Stress evaluation in seven-wire strands based on singular value feature of ultrasonic guided waves. **Structural Health Monitoring**, SAGE Publications Sage UK: London, England, 2021. Citado na p. 16.
- JIA, J.; LI, Y. Deep learning for structural health monitoring: Data, algorithms, applications, challenges, and trends. **Sensors**, v. 23, n. 21, 2023. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/23/21/8824>. Citado na p. 16.
- KARAB, B. Inspection and testing of respirators and anaesthesia machines. Citado na p. 47.
- KIRATIWUDHIKUL, P.; CHANYAGORN, P. Gas mixture control system for oxygen therapy in pre-term infants. IEEE, p. 0–5, 2014. Citado na p. 47.
- KRAUTKRÄMER, J.; KRAUTKRÄMER, H. **Ultrasonic Testing of Materials**. [S.l.]: Springer, 1990. Citado na p. 23.
- KUBRUSLY, A. C.; PÉREZ, N.; OLIVEIRA, T. F.; ADAMOWSKI, J. C.; BRAGA, A. M. B.; WEID, J. P. Von de. Mechanical strain sensing by broadband time reversal in plates. **IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control**, 2016. Citado nas pp. 10, 21, 24, 25 e 28.
- LEE, L. H.; RAJKUMAR, R.; LO, L. H.; WAN, C. H.; ISA, D. Oil and gas pipeline failure prediction system using long range ultrasonic transducers and euclidean-support vector machines classification approach. **Expert Systems with Applications**, v. 40, n. 6, p. 1925 – 1934, 2013. Cited by: 59. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-8487284988&doi=10.1016%2fj.eswa.2012.10.006&partnerID=40&md5=1068d35149d8ca349621678e2ec13429>. Citado na p. 42.
- LEI, Y. **Medical Ventilator System Basics: A clinical guide**. Oxford University Press, 2017. ISBN 9780191827181. Disponível em: <https://oxfordmedicine.com/view/10.1093/med/9780198784975.001.0001/med-9780198784975>. Citado na p. 46.
- LIM, H.; SOHN, H. Online stress monitoring technique based on lamb-wave measurements and a convolutional neural network under static and dynamic loadings. **Experimental Mechanics**, Springer, 2020. Citado na p. 17.

- LIVADIOTIS, S.; EBRAHIMKHANLOU, A.; SALAMONE, S. A helical-based ultrasonic imaging algorithm for structural health monitoring of cylindrical structures. In: . [s.n.], 2019. v. 10972. Cited by: 5. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85067111856&doi=10.11117%2f12.2514308&partnerID=40&md5=f3f3a0d0b7fd2fcf0a84808146718d5a>. Citado na p. 42.
- LJUNG, L. **System Identification: Theory for the User.** [S.l.]: Prentice Hall, 1999. Citado na p. 32.
- LOPES, M. V. de S. **Extração de Características Utilizando Análise de Componentes Independentes para Spike Sorting.** Dissertação (Dissertação de Mestrado) — Universidade Federal do Maranhão, São Luís, Brasil, 2013. Disponível em: <http://tedebc.ufma.br:8080/jspui/handle/tede/1894>. Citado na p. 26.
- LUEPSCHEN, H.; ZHU, L.; LEONHARDT, S. Robust closed-loop control of the inspired fraction of oxygen for the online assessment of recruitment maneuvers. In: . [S.l.: s.n.], 2007. ISSN 05891019. Citado nas pp. 46 e 48.
- MARCU, T.; MIREA, L.; FRANK, P. M.; KOCHS, H. D. System identification and fault diagnosis using dynamic functional-link neural networks. In: **2001 European Control Conference (ECC).** [S.l.: s.n.], 2001. p. 1618–1623. Citado na p. 30.
- MARTIN, R.; OGARKO, V.; KOMATITSCH, D.; JESSELL, M. Parallel three-dimensional electrical capacitance data imaging using a nonlinear inversion algorithm and lp norm-based model regularization. **Measurement: Journal of the International Measurement Confederation**, v. 128, p. 428 – 445, 2018. Cited by: 17. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85049483981&doi=10.1016%2fj.measurement.2018.05.099&partnerID=40&md5=72dfd80dfffd7c05fd7771184af39d5b>. Citado na p. 41.
- MASSARONI, C.; CAPONERO, M. A.; D'AMATO, R.; PRESTI, D. L.; SCHENA, E. Fiber bragg grating measuring system for simultaneous monitoring of temperature and humidity in mechanical ventilation. **Sensors**, v. 17, n. 4, 2017. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/17/4/749>. Citado nas pp. 44 e 46.
- MENDES, D. A.; REVES, G.; PASTRANA, M. A.; DOMINGUES, P. H.; AYALA, H. V. H.; KUBRUSLY, A. C.; MUÑOZ, D. M.; LLANOS, C. H. A comparative analysis of hdl and hls for accelerating machine learning based strain estimation with ultrasonic guided waves. In: **2023 XIII Brazilian Symposium on Computing Systems Engineering (SBESC).** [S.l.: s.n.], 2023. p. 1–6. Citado nas pp. 10, 21, 25, 49 e 50.
- MENDES, D. A.; REVES, G.; PASTRANA, M. A.; DOMINGUES, P. H.; AYALA, H. V. H.; KUBRUSLY, A. C.; MUÑOZ, D. M.; LLANOS, C. H. A comparative analysis of HDL and HLS for accelerating machine learning based strain estimation with ultrasonic guided waves. In: **2023 XIII Brazilian Symposium on Computing Systems Engineering (SBESC).** [S.l.: s.n.], 2023. p. 1–6. Citado na p. 19.

- MEŠIĆ, S.; BABUŠKA, R.; HOOGSTEDEN, H. C.; VERBRAAK, A. F. Computer-controlled mechanical simulation of the artificially ventilated human respiratory system. **IEEE Transactions on Biomedical Engineering**, v. 50, 2003. ISSN 00189294. Citado nas pp. 46 e 48.
- MUÑOZ, D. M.; SANCHEZ, D. F.; LLANOS, C. H.; AYALA-RINCÓN, M. FPGA based floating-point library for CORDIC algorithms. In: **2010 VI Southern Programmable Logic Conference (SPL)**. Porto de Galinhas, Brazil: IEEE, 2010. Citado na p. 50.
- MUÑOZ, D. M.; SANCHEZ, D. F.; LLANOS, C. H.; AYALA-RINCÓN, M. Tradeoff of FPGA design of a floating-point library for arithmetic operators. **Journal of Integrated Circuits and Systems**, Journal of Integrated Circuits and Systems, 2010. Citado na p. 50.
- MUSHIN, W. W. A / w. 1954. Citado na p. 47.
- MUÑOZ, F.; ACUÑA, G. Time series forecasting using narx and narmax models with shallow and deep neural networks. In: **2021 IEEE Latin American Conference on Computational Intelligence (LA-CCI)**. [S.l.: s.n.], 2021. p. 1–6. Citado na p. 31.
- NIU, X.; CHEN, H.-P.; MARQUES, H. R. Piezoelectric transducer array optimization through simulation techniques for guided wave testing of cylindrical structures. In: . [s.n.], 2017. v. 2017-January, p. 424 – 435. Cited by: 9. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85045412642&partnerID=40&md5=7dd5d10cd8131aeb1329e443f594d4c7>. Citado na p. 41.
- PAO, Y.; SACHSE, W.; FUKUOKA, H. Acoustoelasticity and ultrasonic measurements of residual stresses. **Phys. Acoust.**, 1984. Citado na p. 16.
- PARK, J.-H.; KIM, D.-K.; KIM, H.-j.; SONG, S.-J.; CHO, S.-H. Development of ema transducer for inspection of pipelines. **Journal of Mechanical Science and Technology**, v. 31, n. 11, p. 5209 – 5218, 2017. Cited by: 6. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85034973833&doi=10.1007%2fs12206-017-1014-3&partnerID=40&md5=30d16951bb5943e72b02f480dd956d96>. Citado na p. 41.
- POPP, H.; KOLLER, M.; JAHN, M.; BERGMANN, A. Mechanical methods for state determination of lithium-ion secondary batteries: A review. **Journal of Energy Storage**, v. 32, 2020. Cited by: 81; All Open Access, Hybrid Gold Open Access. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85091340815&doi=10.1016%2fjes.2020.101859&partnerID=40&md5=872cc34745c61fe43ae61ec4f7fa4b88>. Citado na p. 39.
- PROGRAM, U. D. o. E. Pacific Northwest National Laboratory for the F. E. M. **Operations & Maintenance Best Practices: A Guide to Achieving Operational Efficiency**. 2010. Published by the U.S. Department of Energy. Citado na p. 16.
- RADZI, M. F. Pid-based fraction of inspired oxygen (fio 2) control system in icu ventilator. **2021 IEEE 7th International Conference on Smart Instrumentation, Measu-**

- rement and Applications (ICSIMA)**, IEEE, p. 185–189, 2021. Citado nas pp. 47 e 48.
- RIVAZ, H.; BOCTOR, E. M.; CHOTI, M. A.; HAGER, G. D. Ultrasound elastography using multiple images. **Medical Image Analysis**, v. 18, n. 2, p. 314 – 329, 2014. Cited by: 48. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84890843168&doi=10.1016%2fj.media.2013.11.002&partnerID=40&md5=f5ef6a12511583d454fbb24d07e5a5d7>. Citado na p. 40.
- ROBERT, R.; MICHEAU, P.; AVOINE, O.; BEAUDRY, B.; BEAULIEU, A.; WALTI, H. A regulator for pressure-controlled total-liquid ventilation. **IEEE Transactions on Biomedical Engineering**, v. 57, n. 9, p. 2267–2276, 2010. Citado nas pp. 44, 46 e 48.
- RODRIGUES, J. J. P. C. Air – oxygen blenders for mechanical ventilators : A literature review. p. 1–15, 2022. Citado na p. 47.
- ROGERS, M.; LO, T.; MALINOWSKI, T.; AFUWAPE, S.; LUM, M.; GRUNDL, B.; TERRY, M. Closed-loop control of the inspired fraction of oxygen in mechanical ventilation. p. 367–376, 2003. Citado nas pp. 47 e 48.
- ROSE, J. L. **Ultrasonic Guided Waves in Solid Media**. [S.l.]: Cambridge University Press, 2014. Citado na p. 24.
- SAMPAIO, R.; AYALA, H.; MUÑOZ, D. M.; LLANOS, C.; COELHO, L.; JACOBI, R. Nonlinear model predictive control hardware implementation with custom-precision floating point operations. **24th Mediterranean Conference on Control and Automation (MED)**, p. pp. 135–140, June 2016. Citado nas pp. 11 e 68.
- SAYIN, F. S. Design , modelling , prototyping and closed loop control of a mechanical ventilator for newborn babies. IEEE, p. 25–27, 2018. Citado nas pp. 47 e 48.
- SEN, D.; AGHAZADEH, A.; MOUSAVI, A.; NAGARAJAIAH, S.; BARANIUK, R.; DABAK, A. Data-driven semi-supervised and supervised learning algorithms for health monitoring of pipes. **Mechanical Systems and Signal Processing**, v. 131, p. 524 – 537, 2019. Cited by: 67. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85067035366&doi=10.1016%2fj.ymssp.2019.06.003&partnerID=40&md5=1ca059235835912f9d53630807ce9384>. Citado na p. 42.
- SHALOO, M.; SCHNALL, M.; KLEIN, T.; HUBER, N.; REITINGER, B. A review of non-destructive testing (NDT) techniques for defect detection: application to fusion welding and future wire arc additive manufacturing processes. **Materials**, MDPI, v. 15, n. 10, p. 3697, 2022. Citado na p. 17.
- SHAN, S.; HOU, Z. Neural network narmax model based unmanned aircraft control surface reconfiguration. In: **2016 9th International Symposium on Computational Intelligence and Design (ISCID)**. [S.l.: s.n.], 2016. v. 1, p. 154–157. Citado na p. 31.

- SHARIFI, R.; CHBAT, N. Dynamic modeling of patient breathing effort in response to change in pressure support in mechanical ventilation. In: **2015 8th Biomedical Engineering International Conference (BMEiCON)**. [S.l.: s.n.], 2015. p. 1–4. Citado na p. 18.
- SINGLA, P.; SUBBARAO, K.; JUNKINS, J. L. Direction-dependent learning approach for radial basis function networks. **IEEE Transactions on Neural Networks**, v. 18, n. 1, p. 203–222, 2007. Citado na p. 31.
- SMITH, I. E.; SHNEERSON, J. M. A laboratory comparison of four positive pressure ventilators used in the home. **European Respiratory Journal**, v. 9, p. 2410–2415, 1996. ISSN 09031936. Citado na p. 47.
- STANZL-TSCHEGG, S.; MEISCHEL, M.; ARCARI, A.; IYYER, N.; APETRE, N.; PHAN, N. Combined cycle fatigue of 7075 aluminum alloy – fracture surface characterization and short crack propagation. **International Journal of Fatigue**, v. 91, p. 352 – 362, 2016. Cited by: 39. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84949666915&doi=10.1016%2fijfatigue.2015.10.022&partnerID=40&md5=fa81430f080ef980a374df21bf4c50b3>. Citado na p. 39.
- TEHRANI, F. T. Automatic control of mechanical ventilation. part 1: Theory and history of the technology. **Journal of Clinical Monitoring and Computing**, v. 22, 2008. ISSN 13871307. Citado na p. 46.
- TEHRANI, F. T. Automatic control of mechanical ventilation. part 2: The existing techniques and future trends. **Journal of Clinical Monitoring and Computing**, v. 22, 2008. ISSN 13871307. Citado na p. 47.
- TEHRANI, F. T. A control system for mechanical ventilation of passive and active subjects. **Computer Methods and Programs in Biomedicine**, v. 110, 2013. ISSN 01692607. Citado na p. 46.
- TEHRANI, F. T.; ROGERS, M.; LO, T.; MALINOWSKI, T.; AFUWAPE, S.; LUM, M.; GRUNDL, B.; TERRY, M. A dual closed-loop control system for mechanical ventilation. **Journal of Clinical Monitoring and Computing**, v. 18, 2004. ISSN 13871307. Citado nas pp. 47 e 48.
- TEIMOURZADEH, H.; MORADZADEH, A.; SHOARAN, M.; MOHAMMADI-IVATLOO, B.; RAZZAGHI, R. High impedance single-phase faults diagnosis in transmission lines via deep reinforcement learning of transfer functions. **IEEE Access**, v. 9, p. 15796–15809, 2021. Citado na p. 30.
- THON, A.; PAINCHAUD-APRIL, G.; DUFF, A. L.; BÉLANGER, P. Optimization of a lorentz forces emat for the reconstruction of the circumferential thickness profile of a steel pipe using high order shear horizontal modes. **NDT and E International**, v. 128, 2022. Cited by: 17. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85125527205&doi=10.1016%2fj.ndte.2022.10.001&partnerID=40&md5=fa81430f080ef980a374df21bf4c50b3>.

- ndteint.2022.102631&partnerID=40&md5=a31fc0278b74b0ac18e1fc9412bb6e28.
Citado na p. 41.
- TONUSSI, G. R. V.; QUINTERO, C. H. L. **Identificação de sistema (ECG x respiração) para detecção de batimentos ectópicos de maneira automática usando rede neural em um SoC**. 2020. Disponível em: <https://bdm.unb.br/handle/10483/30560>.
Citado nas pp. 10, 20, 32 e 68.
- TRAN, A. S.; NGO, H. Q. T.; DONG, V. K.; VO, A. H. Design, control, modeling, and simulation of mechanical ventilator for respiratory support. **Mathematical Problems in Engineering**, v. 2021, n. 1, p. 2499804, 2021. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1155/2021/2499804>. Citado na p. 18.
- VIEIRA, R. B.; LAMBROS, J. Machine learning neural-network predictions for grain-boundary strain accumulation in a polycrystalline metal. **Experimental Mechanics**, Springer Science and Business Media LLC, 2021. Citado na p. 17.
- VIRKKUNEN, I.; KOSKINEN, T.; JESSEN-JUHLER, O.; RINTA-AHO, J. Augmented Ultrasonic Data for Machine Learning. **Journal of Nondestructive Evaluation**, Springer US, v. 40, n. 1, p. 1–11, 2021. ISSN 15734862. Citado nas pp. 18, 21, 26, 28 e 70.
- WANG, X.; FENG, E.; JIANG, C. A microplasticity evaluation method in very high cycle fatigue. **International Journal of Fatigue**, v. 94, p. 6 – 15, 2017. Cited by: 37; All Open Access, Bronze Open Access. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84987920116&doi=10.1016%2fijfatigue.2016.09.004&partnerID=40&md5=9c6b84b86d20dd9621017eccfd7252e8>.
Citado na p. 39.
- WITTEK, A.; DERWICH, W.; KARATOLIOS, K.; FRITZEN, C. P.; VOGT, S.; SCHMITZ-RIXEN, T.; BLASE, C. A finite element updating approach for identification of the anisotropic hyperelastic properties of normal and diseased aortic walls from 4d ultrasound strain imaging. **Journal of the Mechanical Behavior of Biomedical Materials**, v. 58, p. 122 – 138, 2016. Cited by: 55. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84957009275&doi=10.1016%2fj.jmbbm.2015.09.022&partnerID=40&md5=4af6a4fe3ea57b43edfa287716b65baa>.
Citado na p. 39.
- XU, C.; JIANG, S.; LUO, G.; SUN, G.; AN, N.; HUANG, G.; LIU, X. The case for FPGA-Based edge computing. **IEEE Transactions on Mobile Computing**, v. 21, n. 7, p. 2610–2619, 2022. Citado na p. 19.
- XU, Y.; LUO, M.; LIU, Q.; DU, G.; SONG, G. Pzt transducer array enabled pipeline defect locating based on time-reversal method and matching pursuit de-noising. **Smart Materials and Structures**, v. 28, n. 7, 2019. Cited by: 56. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85068639355&doi=10.1088%2f1361-665X%2fab1cc9&partnerID=40&md5=9110313f1a2c69c3f06e13c9916bd17a>. Citado na p. 42.

- YANG, Z.; YANG, H.; TIAN, T.; DENG, D.; HU, M.; MA, J.; GAO, D.; ZHANG, J.; MA, S.; YANG, L.; XU, H.; WU, Z. A review in guided-ultrasonic-wave-based structural health monitoring: From fundamental theory to machine learning techniques. *Ultrasoundics*, 2023. Citado na p. 16.
- YING, Y.; GARRETT, J. H.; OPPENHEIM, I. J.; SOIBELMAN, L.; HARLEY, J. B.; SHI, J.; JIN, Y. Toward data-driven structural health monitoring: Application of machine learning and signal processing to damage detection. *Journal of Computing in Civil Engineering*, v. 27, n. 6, p. 667 – 680, 2013. Cited by: 127. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84886500913&doi=10.1061%2f%28ASCE%29CP.1943-5487.0000258&partnerID=40&md5=a2924e14c0cf828881bb4ed45a013bde>. Citado na p. 42.
- ZHANG, Z.; TANG, F.; CAO, Q.; PAN, H.; WANG, X.; LIN, Z. Deep learning-enriched stress level identification of pretensioned rods via guided wave approaches. *Buildings*, MDPI, 2022. Citado na p. 17.

Anexo A – Trabalhos Publicados

- **Título:** A Comparative Analysis of HDL and HLS for Accelerating Machine Learning based Strain Estimation with Ultrasonic Guided Waves
- **Autores:** Mendes, Davi A. e Reves, Gabriel e Pastrana, M. A. e Domingues, Pedro H. e Ayala, Helon V. H. e Kubrusly, Alan C. e Muñoz, Daniel M. and Llanos, Carlos H.
- **Status:** Publicado
- **DOI:** 10.1109/SBESC60926.2023.10324053
- **Revista/Conferência:** 2023 XIII Brazilian Symposium on Computing Systems Engineering (SBESC)
- **Resumo:** São amplamente utilizadas para identificar defeitos e caracterizar materiais. Recentemente, modelos de aprendizado de máquina baseados em dados foram propostos para estimativa de deformação usando modelos rasos e Análise de Componentes Principais (PCA). No entanto, pouco esforço de pesquisa tem sido direcionado ao desenvolvimento de aceleradores de hardware para estimativa de deformação em tempo real. Este estudo apresenta uma nova análise comparativa de implementações de hardware da PCA em um SoC-FPGA de baixo custo, utilizando Síntese de Alto Nível (HLS) e arquiteturas baseadas em HDL. A comparação foi realizada com base em métricas relevantes: ocupação de hardware, latência e eficiência computacional. Além disso, demonstramos uma análise de escalabilidade considerando a representação de ponto flutuante com diferentes larguras de bits e o número de operadores. A arquitetura baseada em HDL proposta foi capaz de alcançar um desempenho semelhante em comparação com a implementação baseada em HLS. As vantagens dos aceleradores de hardware propostos são evidenciadas por suas capacidades de inferência em tempo real, baixo consumo de energia e utilização reduzida de hardware, associadas a baixa latência e elevada eficiência computacional.

- **Título:** System-on-Chip Machine Learning Pipeline Crack Detection Method with Ultrasonic Data
- **Autores:** Guilherme R. B. Ferreira e Gabriel R. V. Tonussi e Daniel M. Muñoz e Carlos H. Llanos e Alan C. Kubrusly e Helon V. H. Ayala
- **Status:** Em revisão
- **Revista/Conferência:** 2025 IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS
- **Resumo:** A inspeção ultrassônica não destrutiva é um método amplamente utilizado para avaliação de danos em juntas soldadas de tubulações. Com os recentes avanços em inteligência artificial, muitos estudos têm proposto soluções baseadas em aprendizado de máquina para avaliação de danos na inspeção ultrassônica de soldas. No entanto, as soluções propostas geralmente empregam modelos de aprendizado profundo sem focar na otimização da eficiência computacional para sistemas embarcados. Dessa forma, este trabalho se concentra no desenvolvimento de um fluxo de trabalho eficiente de aprendizado de máquina para a construção de um sistema de diagnóstico embarcado baseado em uma solução SoC para inspeção ultrassônica de soldagem de tubulações na borda. Primeiramente, é realizada uma comparação extensiva entre arquiteturas de aprendizado raso e profundo, abordando eficiência e desempenho computacional para determinar o melhor compromisso entre precisão e consumo de hardware. O melhor modelo é então implementado em uma plataforma embarcada para avaliar seu desempenho. Os resultados mostram que a solução proposta aumenta o desempenho (precisão superior a 99%) em comparação com modelos de última geração construídos no mesmo conjunto de dados. Além disso, a abordagem proposta também fornece um modelo econômico para implementações em sistemas embarcados. Experimentos em uma plataforma embarcada SoC demonstraram que uma imagem ultrassônica é processada em aproximadamente 33 ms, seguindo o método proposto de co-projeto hardware/software.