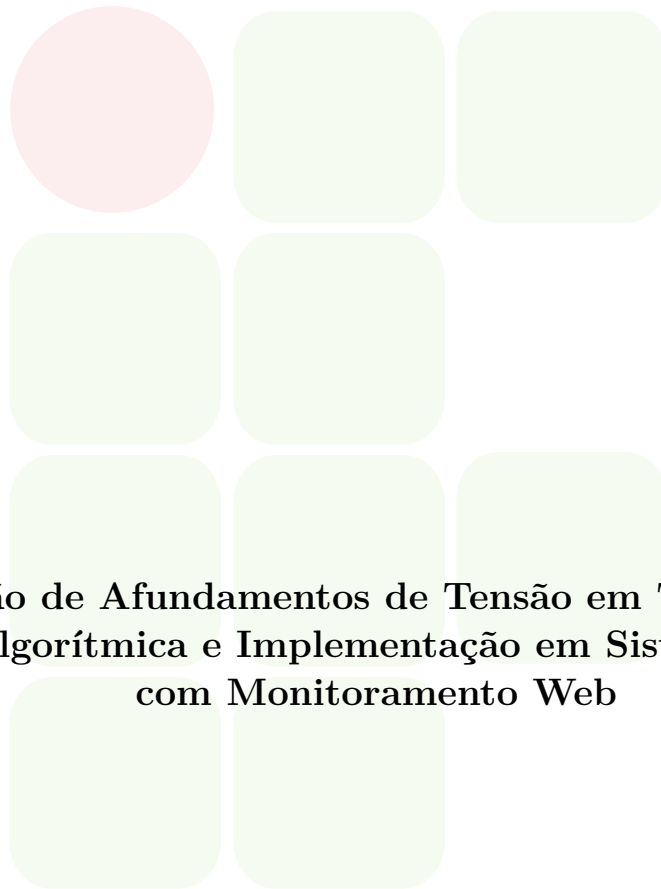


INSTITUTO FEDERAL DA PARAÍBA
COORDENAÇÃO DO CURSO SUPERIOR DE BACHARELADO EM
ENGENHARIA ELÉTRICA

LUCAS MEDEIROS CARNEIRO COSTA



**Detecção de Afundamentos de Tensão em Tempo Real:
Avaliação Algorítmica e Implementação em Sistema Embarcado
com Monitoramento Web**

JOÃO PESSOA - PB

2026

LUCAS MEDEIROS CARNEIRO COSTA

**Detecção de Afundamentos de Tensão em Tempo Real:
Avaliação Algorítmica e Implementação em Sistema Embarcado
com Monitoramento Web**

Trabalho de conclusão de curso submetido à Coordenação do Curso Superior de Bacharelado em Engenharia Elétrica do Instituto Federal da Paraíba, como parte dos requisitos para a obtenção do grau de engenheiro eletricista.

Orientador: Dr. Edgard Luiz Lopes Fabricio

JOÃO PESSOA - PB

2026

Dados Internacionais de Catalogação na Publicação – CIP
Biblioteca Nilo Peçanha – IFPB, *campus* João Pessoa

C837d	<p>Costa, Lucas Medeiros Carneiro. Detecção de afundamentos de tensão em tempo real: avaliação algorítmica e implementação em sistema embarcado com monitoramento Web / Lucas Medeiros Carneiro Costa. – 2026. 66 f.: il. TCC (Graduação em Engenharia Elétrica) – Instituto Federal de Educação, Ciência e Tecnologia da Paraíba – IFPB / Coordenação de Engenharia Elétrica. Orientador: Prof. Dr. Edgard Luiz Lopes Fabricio. 1. Afundamento de tensão. 2. Sistemas embarcados. 3. Qualidade da energia elétrica. 4. Microcontroladores. I. Título. CDU 621.311</p>
-------	--

LUCAS MEDEIROS CARNEIRO COSTA

**DETECÇÃO DE AFUNDAMENTOS DE TENSÃO EM TEMPO REAL: AVALIAÇÃO ALGORÍTIMICA E
IMPLEMENTAÇÃO EM SISTEMA EMBARCADO COM MONITORAMENTO WEB**

TRABALHO DE CONCLUSÃO DE CURSO (TCC) submetido a Coordenação do Curso Superior de Bacharelado em Engenharia Elétrica, do Instituto Federal da Paraíba (IFPB), como parte dos requisitos institucionais para a obtenção do Título de **ENGENHEIRO ELETRICISTA..**

Aprovado em 25/06/2026 - 14:00

Membros da Banca Examinadora:

Prof. Dr.

EDGARD LUIZ LOPES FABRICIO

Instituto Federal da Paraíba
Orientador Interno

Prof. Dr. ALVARO DE MEDEIROS MACIEL

Instituto Federal da Paraíba
Examinador Interno

Prof. Dr. CICERO ALISSON DOS SANTOS

Instituto Federal da Paraíba
Examinador Interno

Documento assinado eletronicamente por:

- **Edgard Luiz Lopes Fabricio**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 03/07/2026 10:57:32.
- **Alvaro de Medeiros Maciel**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 03/07/2026 11:27:07.
- **Cicero Alisson dos Santos**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 03/07/2026 15:47:47.

Este documento foi emitido pelo SUAP em 03/07/2026. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código 899490

Verificador: 164048ca16

Código de Autenticação:



AGRADECIMENTOS

Agradeço a Deus por me conceder saúde, força e perseverança ao longo do período de realização deste trabalho.

À minha família, em especial à minha mãe e aos meus irmãos, por proporcionarem todo o apoio e a motivação ao longo do período de realização deste curso.

À Laissa Kerli, por ter me dado forças nos momentos de dificuldade e constantes incentivos para que eu prosseguisse até a conclusão deste trabalho.

Aos amigos, em especial Vinicius Simão, João Paulo e Mykaella Fernanda, que estiveram comigo, ajudando e sempre lado a lado diante de qualquer problema ou dificuldade.

Agradeço ao Instituto Federal da Paraíba (IFPB) pela formação acadêmica oferecida e pelas oportunidades proporcionadas ao longo da graduação, que possibilitaram a realização deste trabalho de conclusão de curso.

Agradeço ao Programa de Educação Tutorial (PET) e ao Laboratório de Acionamentos Controle e Automação (LACA) pelo ambiente de aprendizado constante e pelo incentivo à excelência acadêmica, que foram fundamentais para a realização deste trabalho.

Agradeço ao Polo de Inovação do IFPB pela oportunidade de participação em projetos de pesquisa e desenvolvimento, experiências fundamentais para o meu desenvolvimento técnico, científico e profissional.

Por último, agradeço ao meu orientador, Edgard Luiz, por todo o apoio, incentivo e atenção depositados em mim durante minha trajetória acadêmica. Sua orientação teve início ainda no projeto desenvolvido no Polo de Inovação e se consolidou ao longo deste trabalho de conclusão de curso. Sua dedicação e competência são constantes fontes de inspiração para todos que têm o privilégio de trabalhar ao seu lado.

A todos que, direta ou indiretamente, contribuíram para a minha formação humana e técnica, minha eterna gratidão.

*"A inteligência é o único meio que possuímos para dominar os nossos instintos."
(Sigmund Freud)*

RESUMO

Este trabalho apresenta o desenvolvimento e a implementação prática de um sistema autônomo para detecção de afundamentos de tensão (*voltage sag*) em tempo real, integrando processamento digital de sinais (PDS) com aquisição a 10 kHz e latência de detecção inferior a 3,0 ms e conectividade web. A arquitetura organiza-se em três blocos principais: (i) modelagem e simulação comparativa de cinco algoritmos matemáticos de detecção; (ii) implementação embarcada em um microcontrolador STM32F767, utilizando aquisição determinística a 10 kHz via conversor analógico-digital (ADC), temporizador de *hardware* e acesso direto à memória (DMA); e (iii) registro temporal viabilizado por *Real-Time Clock* (RTC) e exposição de resultados por meio de um servidor HTTP nativo operando com pilha LwIP e *Server Side Includes* (SSI). Um experimento computacional avaliou a latência dos algoritmos frente a diferentes ângulos de incidência da falta (*Point-on-Wave*: 0°, 45° e 90°) e magnitudes de afundamento (0,5 pu e 0,8 pu). Para viabilizar a execução autônoma, o método de melhor desempenho computacional (RMS Móvel de Meio Ciclo) foi adaptado no *firmware* por meio de *buffer* circular e soma em execução (*running sum*), reduzindo a complexidade iterativa de $O(N)$ para $O(1)$. Os resultados indicam que métodos de janela fixa possuem latência não determinística, enquanto algoritmos baseados em pico apresentam perda de sensibilidade quando a falta incide na região de pico da forma de onda (90°). O algoritmo RMS Móvel otimizado obteve fluxo de detecção contínuo, registrando tempo de resposta teórico entre 1,0 ms e 3,0 ms para faltas severas. Ensaios de validação em bancada com osciloscópio mediram uma latência física típica de 1,60 ms por evento, atestando a aplicabilidade do hardware na proteção e monitoramento descentralizado da qualidade de energia.

Palavras-chave: Afundamento de tensão. Processamento digital de sinais. RMS Móvel. Sistema embarcado. Microcontrolador STM32. Qualidade de energia elétrica.

ABSTRACT

This work presents the development and practical implementation of an autonomous system for real-time voltage sag detection, integrating digital signal processing (DSP) with 10 kHz acquisition and detection latency below 3.0 ms and web connectivity. The architecture is organized into three main blocks: (i) modeling and comparative simulation of five mathematical detection algorithms; (ii) embedded implementation on an STM32F767 microcontroller, using deterministic acquisition at 10 kHz via an analog-to-digital converter (ADC), hardware timer, and direct memory access (DMA); and (iii) time logging enabled by a Real-Time Clock (RTC) and results exposure through a native HTTP server operating with the LwIP stack and Server Side Includes (SSI). A computational experiment evaluated the latency of the algorithms under different fault incidence angles (Point-on-Wave: 0° , 45° , and 90°) and sag magnitudes (0.5 pu and 0.8 pu). To enable autonomous execution, the computationally best-performing method (Half-Cycle Moving RMS) was adapted into the firmware using a circular buffer and a running sum, reducing the iterative complexity from $O(N)$ to $O(1)$. Results indicate that fixed-window methods have non-deterministic latency, while peak-based algorithms exhibit a loss of sensitivity when the fault occurs at the peak of the waveform (90°). The optimized Moving RMS algorithm achieved a continuous detection flow, recording a theoretical response time between 1.0 ms and 3.0 ms for severe faults. Bench validation tests with an oscilloscope measured a typical physical latency of 1.60 ms per event, attesting to the hardware's applicability in decentralized power quality protection and monitoring.

Keywords: Voltage sag. Digital signal processing. Moving RMS. Embedded system. STM32 microcontroller. Power quality.

LISTA DE FIGURAS

Figura 1 – Placa de desenvolvimento NUCLEO-F767ZI.	22
Figura 2 – Janela de cálculo do método RMS em bloco utilizando um ciclo completo da forma de onda.	26
Figura 3 – Janela de análise utilizada no método de tensão de pico em meio ciclo.	27
Figura 4 – Deslocamento da janela de cálculo no método RMS móvel.	28
Figura 5 – Diagrama de blocos funcional da implementação embarcada.	32
Figura 6 – Sinal de tensão com inserção de afundamento em 0° (cruzamento por zero).	35
Figura 7 – Comportamento da detecção utilizando janelas de RMS Móvel (Cenário 0°).	35
Figura 8 – Comportamento da detecção utilizando métodos de Tensão de Pico (Cenário 0°).	36
Figura 9 – Comportamento da detecção utilizando cálculo RMS em Bloco vs Pico em Bloco (Cenário 0°).	36
Figura 10 – Sinalização (<i>flags</i>) de detecção dos algoritmos (Cenário 0°).	37
Figura 11 – Atraso de detecção para cada algoritmo no Cenário 1 (0°).	38
Figura 12 – Sinal de tensão com inserção de afundamento em 45°.	39
Figura 13 – Comportamento da detecção utilizando janelas de RMS Móvel (Cenário 45°).	39
Figura 14 – Comportamento da detecção utilizando métodos de Tensão de Pico (Cenário 45°).	40
Figura 15 – Comportamento da detecção utilizando cálculo RMS em Bloco vs Pico em Bloco (Cenário 45°).	40
Figura 16 – Sinalização (<i>flags</i>) de detecção dos algoritmos (Cenário 45°).	41
Figura 17 – Atraso de detecção para cada algoritmo no Cenário 2 (45°).	42
Figura 18 – Sinal de tensão com inserção de afundamento em 90° (pico positivo).	43
Figura 19 – Comportamento da detecção utilizando métodos de Tensão de Pico (Cenário 90°).	43
Figura 20 – Comportamento da detecção utilizando janelas de RMS Móvel (Cenário 90°).	44
Figura 21 – Comportamento da detecção utilizando cálculo RMS em Bloco vs Pico em Bloco (Cenário 90°).	44
Figura 22 – Sinalização (<i>flags</i>) de detecção dos algoritmos (Cenário 90°).	45
Figura 23 – Atraso de detecção para cada algoritmo no Cenário 3 (90°).	46
Figura 24 – Validação prática do tempo de detecção utilizando osciloscópio digital.	50
Figura 25 – Interface web desenvolvida para monitoramento do histórico de afundamentos de tensão.	52

LISTA DE TABELAS

Tabela 1 – Comparativo entre trabalhos relacionados e o presente trabalho	24
Tabela 2 – Resumo do atraso de detecção no Cenário 1 (0°).	38
Tabela 3 – Resumo do atraso de detecção no Cenário 2 (45°).	42
Tabela 4 – Resumo do atraso de detecção no Cenário 3 (90°).	46
Tabela 5 – Atraso de detecção para afundamento leve (0,8 pu) no cruzamento por zero (0°).	47

LISTA DE ABREVIATURAS E SIGLAS

ADC	<i>Analog-to-Digital Converter</i> (Conversor Analógico-Digital)
ALU	<i>Arithmetic Logic Unit</i> (Unidade Lógica e Aritmética)
ANEEL	<i>Agência Nacional de Energia Elétrica</i>
ARM	<i>Advanced RISC Machine</i>
CPU	<i>Central Processing Unit</i> (Unidade Central de Processamento)
DHCP	<i>Dynamic Host Configuration Protocol</i>
DMA	<i>Direct Memory Access</i> (Acesso Direto à Memória)
DSP	<i>Digital Signal Processing</i> (Processamento Digital de Sinais)
FPU	<i>Floating-Point Unit</i> (Unidade de Ponto Flutuante)
GPIO	<i>General Purpose Input/Output</i> (Pino Digital)
HAL	<i>Hardware Abstraction Layer</i> (Camada de Abstração de Hardware)
HTML	<i>HyperText Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IDE	<i>Integrated Development Environment</i> (Ambiente de Desenvolvimento Integrado)
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IP	<i>Internet Protocol</i>
LwIP	<i>Lightweight IP</i>
MQTT	<i>Message Queuing Telemetry Transport</i>
PRODIST	<i>Procedimentos de Distribuição de Energia Elétrica no Sistema Elétrico Nacional</i>
pu	<i>Per unit</i> (Por unidade)
QEE	<i>Qualidade da Energia Elétrica</i>
RAM	<i>Random Access Memory</i>
RMS	<i>Root Mean Square</i> (Valor Eficaz)

RTC	<i>Real-Time Clock</i>
SSI	<i>Server Side Includes</i>
TCP	<i>Transmission Control Protocol</i>
THD	<i>Total Harmonic Distortion</i> (Distorção Harmônica Total)

SUMÁRIO

1	INTRODUÇÃO	15
1.1	JUSTIFICATIVA	16
1.2	OBJETIVOS	17
1.2.1	Objetivo Geral	17
1.2.2	Objetivos Específicos	17
1.3	ESTRUTURA DO TRABALHO	17
2	FUNDAMENTAÇÃO TEÓRICA	18
2.1	QUALIDADE DA ENERGIA ELÉTRICA	18
2.1.1	Afundamentos de Tensão (<i>Voltage Sags</i>)	18
2.2	PROCESSAMENTO DIGITAL DE SINAIS APLICADO À QEE	19
2.2.1	Amostragem e o Teorema de Nyquist-Shannon	19
2.2.2	Representação Discreta do Sinal	19
2.2.3	Janelamento de Dados	20
2.3	ARQUITETURA DE SISTEMAS EMBARCADOS PARA TEMPO REAL	20
2.3.1	Microcontroladores da Família STM32	21
2.3.2	Conectividade em Sistemas Embarcados	22
2.3.3	Ambiente de Desenvolvimento STM32CubeIDE	23
2.4	TRABALHOS RELACIONADOS	23
3	METODOLOGIA	25
3.1	VISÃO GERAL DA METODOLOGIA	25
3.2	MÉTODOS PARA DETECÇÃO DE AFUNDAMENTOS DE TENSÃO	25
3.2.1	Método RMS em Bloco	25
3.2.2	Tensão de Pico em um Ciclo Completo	26
3.2.3	Tensão de Pico em Meio Ciclo	27
3.2.4	Método RMS Móvel com Janela de Um Ciclo	27
3.2.5	Método RMS Móvel com Janela de Meio Ciclo	28
3.3	SIMULAÇÃO E AVALIAÇÃO DOS ALGORITMOS	28
3.3.1	Critério e registros adotados	29
3.4	SISTEMA DE AQUISIÇÃO DO SINAL	29

3.4.1	Frequência do sinal monitorado	30
3.4.2	Taxa de amostragem	30
3.4.3	Geração do sinal de teste	30
3.4.4	Aquisição com temporizador e DMA	31
3.5	IMPLEMENTAÇÃO EMBARCADA	31
3.6	INTERFACE WEB PARA MONITORAMENTO	32
3.7	SÍNTESE DA METODOLOGIA	33
4	RESULTADOS	34
4.1	AVALIAÇÃO DOS MÉTODOS DE DETECÇÃO EM SIMULAÇÃO	34
4.1.1	Cenário 1: Início do afundamento em 0 graus	34
4.1.2	Cenário 2: Início do afundamento em 45 graus	38
4.1.3	Cenário 3: Início do afundamento em 90 graus	42
4.1.4	Efeito da magnitude do afundamento na latência de detecção	46
4.1.5	Síntese da validação em simulação	47
4.2	IMPLEMENTAÇÃO EM SISTEMA EMBARCADO	47
4.2.1	Otimização computacional do RMS Móvel	47
4.2.2	Configuração de hardware e processamento em tempo real	48
4.2.3	Registro de eventos com RTC	49
4.2.4	Validação experimental em hardware	49
4.3	VISUALIZAÇÃO DOS EVENTOS DETECTADOS EM INTERFACE WEB	50
5	DISCUSSÃO	53
5.1	INFLUÊNCIA DA NATUREZA MATEMÁTICA, ÂNGULO DE INCIDÊNCIA E MAGNITUDE	53
5.2	CONSIDERAÇÕES SOBRE O DESEMPENHO EM TEMPO REAL	54
5.3	VIABILIDADE AUTÔNOMA E CONECTIVIDADE	55
6	CONCLUSÃO	56
	REFERÊNCIAS	58
	ANEXO A – ANÁLISE DE AFUNDAMENTOS	60
	ANEXO B – INTERFACE WEB (STM32)	64

1 INTRODUÇÃO

A qualidade da energia elétrica (QEE) constitui um fator fundamental para o funcionamento adequado de sistemas elétricos modernos, influenciando diretamente o desempenho, a confiabilidade e a vida útil dos equipamentos conectados à rede (Zhang; Yan, 2020). Com o avanço da Indústria 4.0 e a crescente digitalização dos processos industriais, observa-se um aumento significativo na utilização de dispositivos eletrônicos sensíveis, tais como controladores digitais, inversores de frequência e sistemas de automação baseados em microprocessadores, além de dispositivos IoT integrados à infraestrutura produtiva (Santos, 2020). Embora essenciais para a modernização industrial, esses dispositivos apresentam elevada suscetibilidade a distúrbios transitórios na rede elétrica, tais como afundamentos de tensão (*voltage sags*), interrupções momentâneas e sobretensões impulsivas, que podem ocasionar falhas operacionais, reinicializações indesejadas ou até danos aos equipamentos.

Os impactos econômicos associados a problemas de QEE são expressivos. De acordo com (Pinkerton, 2025; Mazurkiewicz, 2001), distúrbios na qualidade da energia elétrica e interrupções no fornecimento podem resultar em prejuízos superiores a 150 bilhões de dólares anuais na economia dos Estados Unidos. No contexto industrial brasileiro, (SBQEE, 2003) indicam que tais prejuízos estão diretamente relacionados à crescente sensibilidade das cargas modernas, sendo que aproximadamente 15% das perdas de matéria-prima durante interrupções ocorrem nos primeiros instantes do evento. Adicionalmente, a crescente inserção de fontes renováveis nos sistemas elétricos introduz novos desafios à QEE, motivando o desenvolvimento do conceito de *energy quality*, voltado à análise das variações dinâmicas dos fluxos de potência ao longo do tempo (Zhang; Yan, 2020).

Dentre os fenômenos associados à QEE, destacam-se os afundamentos de tensão, caracterizados por reduções momentâneas no valor eficaz da tensão, com duração variando de poucos ciclos a alguns minutos (Deckmann; Pomilio, 2024). Esses eventos figuram entre as perturbações mais frequentes e impactantes em sistemas elétricos (Radhakrishna; Eshwardas; Chebiyam, 2001), podendo ocasionar desde falhas intermitentes até a interrupção completa de processos industriais automatizados. Em aplicações de controle em tempo real, mesmo distúrbios de curta duração podem resultar em danos operacionais e prejuízos financeiros significativos (SBQEE, 2003).

Nesse contexto, a detecção rápida e confiável de afundamentos de tensão torna-se um requisito essencial para sistemas de monitoramento e proteção. Métodos tradicionais de detecção, embora amplamente utilizados, podem apresentar limitações relacionadas ao tempo de resposta, especialmente devido à dependência de janelas de processamento fixas ou do instante de ocorrência da falta (Martins; Couto; Afonso, 2003). À medida que a sensibilidade das cargas aumenta, cresce também a necessidade de técnicas capazes de identificar distúrbios com baixa latência e comportamento determinístico.

Diante dessas limitações, técnicas de processamento digital de sinais (PDS) têm sido amplamente exploradas com o objetivo de equilibrar precisão, robustez e velocidade de resposta (Marafão, 2004). Métodos baseados no cálculo do valor eficaz (RMS) e na análise de valores de pico representam abordagens consolidadas; entretanto, sua eficiência em aplicações em tempo real depende diretamente da estratégia de implementação e da arquitetura de processamento adotada (Naidoo; Pillay, 2007). Nesse cenário, o avanço dos microcontroladores de alto desempenho viabiliza a execução de algoritmos complexos em hardware dedicado, possibilitando o desenvolvimento de sistemas compactos, de baixo custo e com elevada capacidade de processamento.

Com base nessa análise, este trabalho propõe a avaliação comparativa de diferentes métodos matemáticos para a detecção de afundamentos de tensão, com ênfase na influência do ângulo de incidência da falta (*Point-on-Wave*) e da severidade do distúrbio sobre o tempo de resposta dos algoritmos. A partir dessa análise, realiza-se a implementação do método mais eficiente em um sistema embarcado baseado no microcontrolador STM32F767, visando validar sua aplicabilidade em um contexto de monitoramento de qualidade de energia em tempo real.

1.1 JUSTIFICATIVA

Apesar da ampla disponibilidade de métodos para análise de qualidade de energia, muitos apresentam limitações quando aplicados a sistemas que demandam resposta em tempo real. Em particular, técnicas tradicionais de detecção de afundamentos de tensão podem introduzir atrasos dependentes da janela de processamento ou do instante de ocorrência da falta, comprometendo seu desempenho em cenários com cargas altamente sensíveis (Radhakrishna; Eshwardas; Chebiyam, 2001).

Nesse contexto, torna-se necessário investigar abordagens que apresentem baixa latência e comportamento determinístico, independentemente das condições instantâneas do sinal. A análise integrada da influência do ângulo de incidência da falta e da magnitude do afundamento sobre o desempenho dos algoritmos ainda é pouco explorada na literatura, especialmente quando associada à implementação em sistemas embarcados (Deckmann; Pomilio, 2024).

Adicionalmente, a transposição de modelos teóricos para sistemas físicos reais impõe desafios relacionados às limitações de recursos computacionais, temporização e confiabilidade da aquisição de dados. Assim, o desenvolvimento de uma solução otimizada, que integre aquisição determinística, processamento eficiente e capacidade de monitoramento remoto, mostra-se relevante tanto do ponto de vista acadêmico quanto aplicado, contribuindo para o avanço de sistemas de monitoramento de qualidade de energia em tempo real (Martins; Couto; Afonso, 2003).

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Avaliar e implementar métodos de detecção de afundamentos de tensão em sistemas elétricos, com foco na minimização da latência de resposta e na viabilidade de aplicação em sistemas embarcados para monitoramento em tempo real.

1.2.2 Objetivos Específicos

Para o alcance do objetivo geral, definem-se os seguintes objetivos específicos:

1. implementar algoritmos de detecção baseados em RMS e valores de pico em ambiente de simulação computacional;
2. analisar a influência do ângulo de incidência da falta (*Point-on-Wave*) e da magnitude do afundamento no tempo de latência dos métodos;
3. selecionar e otimizar o algoritmo mais eficiente para execução em tempo real;
4. desenvolver um sistema embarcado para aquisição determinística de sinais e processamento autônomo;
5. validar experimentalmente o desempenho do sistema por meio de ensaios em bancada;
6. integrar o sistema a uma interface web embarcada para visualização remota dos eventos.

1.3 ESTRUTURA DO TRABALHO

Este trabalho está organizado em seis capítulos. O Capítulo 2 apresenta a fundamentação teórica sobre qualidade de energia e técnicas de processamento digital de sinais. O Capítulo 3 descreve a metodologia experimental e a arquitetura do sistema embarcado. O Capítulo 4 apresenta os resultados obtidos em simulação e na validação experimental em hardware. O Capítulo 5 discute o desempenho dos métodos sob diferentes condições de operação, bem como as limitações do sistema. Por fim, o Capítulo 6 apresenta as conclusões e sugestões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os fundamentos teóricos necessários ao desenvolvimento do sistema proposto. São abordados, nesta ordem: a Qualidade da Energia Elétrica (QEE) e os afundamentos de tensão; os princípios de Processamento Digital de Sinais (PDS); os métodos matemáticos utilizados para detecção dos afundamentos; os aspectos de sistemas embarcados e monitoramento remoto; o ambiente de desenvolvimento e o microcontrolador STM32F767 utilizados neste trabalho.

2.1 QUALIDADE DA ENERGIA ELÉTRICA

A Qualidade da Energia Elétrica refere-se ao conjunto de características que definem a adequação da energia fornecida aos equipamentos elétricos, garantindo seu funcionamento correto, seguro e eficiente. Desvios nos parâmetros ideais de tensão, frequência e forma de onda podem resultar em falhas operacionais, redução da vida útil de equipamentos e prejuízos econômicos.

A QEE é regulada por normas técnicas como o PRODIST Módulo 8 da Agência Nacional de Energia Elétrica (ANEEL) (Agência Nacional de Energia Elétrica (ANEEL), 2017) e a norma IEEE 1159 (IEEE. . . , 2019), que classificam os principais distúrbios elétricos, entre os quais se destacam sobretensões, subtensões, interrupções, harmônicos e afundamentos de tensão. Em ambientes industriais e comerciais, a detecção e a caracterização desses eventos desempenham papel fundamental na análise da confiabilidade do sistema elétrico.

2.1.1 Afundamentos de Tensão (*Voltage Sags*)

Os afundamentos de tensão, também conhecidos como *voltage sags*, são caracterizados pela redução temporária da magnitude da tensão eficaz (RMS) para valores entre 0,1 e 0,9 pu da tensão nominal, com duração variando de meio ciclo a alguns segundos (IEEE. . . , 2019; Agência Nacional de Energia Elétrica (ANEEL), 2017). Esse distúrbio é frequentemente causado por curtos-circuitos na rede, partidas de grandes motores, energização de transformadores ou variações abruptas de carga (IEEE. . . , 2019).

Embora sua duração seja relativamente curta, os afundamentos podem causar impactos significativos, como desligamento de equipamentos sensíveis, falhas em processos industriais e perda de dados em sistemas computacionais. A severidade é determinada principalmente por dois fatores: a profundidade (nível de redução da tensão) e a duração do evento (Agência Nacional de Energia Elétrica (ANEEL), 2017). Dessa forma, a detecção rápida e precisa de afundamentos é essencial para sistemas de monitoramento de qualidade de energia.

2.2 PROCESSAMENTO DIGITAL DE SINAIS APLICADO À QEE

O Processamento Digital de Sinais (DSP, *Digital Signal Processing*) é fundamental para a análise de sinais elétricos em sistemas de monitoramento de qualidade de energia (Ribeiro, 2005). A partir da conversão do sinal analógico em dados discretos, aplicam-se algoritmos matemáticos para extração de características como valor eficaz, amplitude e frequência.

2.2.1 Amostragem e o Teorema de Nyquist-Shannon

O processo de digitalização de sinais baseia-se na amostragem, que consiste na aquisição periódica do valor do sinal contínuo em instantes discretos de tempo (Oppenheim; Willsky; Young, 2002). Dado um sinal contínuo $v(t)$, sua versão amostrada é obtida pela multiplicação por um trem de impulsos periódicos $p(t)$, com período de amostragem T_s , conforme a Equação (2.1).

$$v_s(t) = v(t)p(t) = \sum_{n=-\infty}^{+\infty} v(nT_s) \delta(t - nT_s), \quad (2.1)$$

onde $\delta(t - nT_s)$ representa o impulso unitário no instante nT_s , e a frequência de amostragem é definida como:

$$f_s = \frac{1}{T_s} \quad (2.2)$$

O Teorema de Nyquist-Shannon estabelece a condição fundamental para que a amostragem seja realizada sem perda de informação: a frequência f_s deve ser pelo menos o dobro da maior componente de frequência f_{max} presente no sinal (Oppenheim; Willsky; Young, 2002; Lathi, 2005), conforme a Equação (2.3).

$$f_s \geq 2f_{max} \quad (2.3)$$

O não atendimento dessa condição resulta no fenômeno de *aliasing*, no qual componentes acima de $f_s/2$ são erroneamente mapeadas para frequências mais baixas no espectro discreto (Oppenheim; Willsky; Young, 2002), conforme a Equação (2.4).

$$f_{alias} = |f - kf_s|, \quad k \in \mathbf{Z}, \quad (2.4)$$

onde f é a frequência original do componente e k é o inteiro que minimiza $|f - kf_s|$.

2.2.2 Representação Discreta do Sinal

Uma vez satisfeita a condição de Nyquist, o sinal contínuo é representado pela sequência discreta de amostras $v[n]$ (Lathi, 2005):

$$v[n] = v(nT_s), \quad n \in \mathbf{Z} \quad (2.5)$$

A qualidade dessa representação depende da resolução de quantização do conversor analógico-digital. Para um conversor de b bits com tensão de referência V_{ref} , o passo de quantização Δ e o erro máximo ϵ_q são dados por (Oppenheim; Willsky; Young, 2002):

$$\Delta = \frac{V_{ref}}{2^b} \quad (2.6)$$

$$|\epsilon_q| \leq \frac{\Delta}{2} = \frac{V_{ref}}{2^{b+1}} \quad (2.7)$$

Quanto maior o número de bits b , menor o passo de quantização e maior a fidelidade da representação digital.

2.2.3 Janelamento de Dados

O janelamento consiste em selecionar e ponderar um subconjunto finito de amostras do sinal para análise local (Lathi, 2005). Matematicamente, a operação é definida pela multiplicação ponto a ponto do sinal $v[n]$ por uma função de janela $w[n]$ de comprimento N :

$$x_w[n] = v[n]w[n], \quad 0 \leq n \leq N - 1 \quad (2.8)$$

A janela retangular é o caso mais simples, com peso unitário dentro do intervalo:

$$w[n] = \begin{cases} 1, & 0 \leq n \leq N - 1 \\ 0, & \text{caso contrário} \end{cases} \quad (2.9)$$

Em algoritmos de detecção de distúrbios elétricos, o janelamento pode ser aplicado de duas formas. Na abordagem de **bloco fixo**, a janela avança em passos de N amostras, atualizando o resultado apenas ao término de cada bloco. Na abordagem de **janela deslizando**, a janela avança uma amostra por vez, permitindo atualização contínua a cada novo instante de amostragem. Essa diferença tem impacto direto na latência de detecção de eventos transitórios, sendo o princípio matemático subjacente aos métodos avaliados na Seção 3.2.

2.3 ARQUITETURA DE SISTEMAS EMBARCADOS PARA TEMPO REAL

Sistemas embarcados voltados ao monitoramento da qualidade de energia exigem comportamento temporal determinístico, ou seja, com tempo de execução fixo. Em aplicações de detecção de distúrbios como afundamentos de tensão, a precisão na aquisição e

no processamento dos dados é fundamental para garantir respostas rápidas e confiáveis. O uso de microcontroladores de alto desempenho viabiliza a implementação de algoritmos DSP em hardware dedicado, resultando em sistemas compactos e de baixo custo com capacidade de operação em tempo real.

2.3.1 Microcontroladores da Família STM32

Os microcontroladores STM32, desenvolvidos pela STMicroelectronics com base na arquitetura ARM Cortex-M, são amplamente empregados em aplicações de processamento de sinais devido ao seu elevado desempenho, baixo consumo de energia e ampla variedade de periféricos integrados.

O STM32F767 utilizado neste trabalho baseia-se no núcleo ARM Cortex-M7 com frequência de até 216 MHz e conta com unidade de ponto flutuante (FPU) integrada, o que acelera operações como o cálculo de raiz quadrada exigido pelo algoritmo RMS (STM32F76XXX... , 2024). Seus principais periféricos para esta aplicação são os conversores analógico-digitais (ADC), os temporizadores de alta precisão, o controlador de acesso direto à memória (DMA) e o controlador MAC Ethernet.

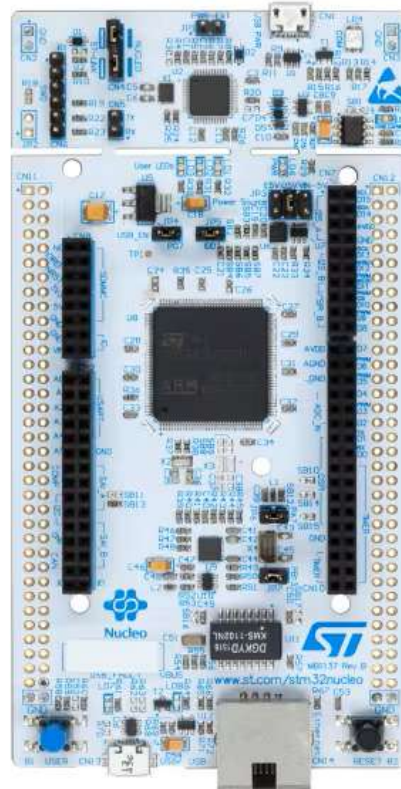
Os temporizadores de hardware são periféricos capazes de gerar eventos periódicos com precisão determinística, independentemente da execução do software (STMicroelectronics, 2016a). No contexto de aquisição de sinais, essa característica é fundamental: ao configurar o temporizador para acionar o ADC em intervalos fixos, garante-se que as amostras sejam coletadas em instantes igualmente espaçados no tempo, condição necessária para a correta estimativa do valor RMS. Caso a amostragem dependesse do fluxo de execução do programa, variações no tempo de execução de outras rotinas introduziriam irregularidades no intervalo entre amostras, comprometendo a precisão do cálculo.

O DMA é um periférico presente em microcontroladores que realiza transferências de dados entre dois endereços de memória, ou entre um periférico e a memória, sem intervenção da CPU (STMicroelectronics, 2016a). Isso permite que operações de movimentação de dados ocorram em paralelo com a execução do programa principal, liberando o processador para outras tarefas. Em aplicações de aquisição de sinais, o DMA é comumente associado ao ADC: a cada conversão concluída, o valor convertido é transferido automaticamente para a memória RAM sem que a CPU precise interromper sua execução para realizar essa leitura. Ao término da transferência, o DMA gera uma interrupção notificando o processador, que passa então a processar a amostra recebida. Sem esse mecanismo, a CPU precisaria aguardar o término de cada conversão antes de prosseguir, o que inviabilizaria o processamento dentro do intervalo de amostragem de 100 μ s adotado neste trabalho.

A plataforma de desenvolvimento utilizada neste trabalho foi a placa NUCLEO-F767ZI apresentada na Figura 1, que integra o microcontrolador STM32F767ZI e oferece

recursos adicionais para prototipagem rápida, como interface de depuração ST-LINK, conectores compatíveis com Arduino e acesso facilitado aos pinos de entrada e saída (STMicroelectronics, 2016a).

Figura 1 – Placa de desenvolvimento NUCLEO-F767ZI.



Fonte: (STMicroelectronics, 2016a).

2.3.2 Conectividade em Sistemas Embarcados

Para viabilizar a comunicação em rede de microcontroladores com recursos restritos, o uso de pilhas de protocolo otimizadas é essencial. O LwIP (*Lightweight IP*) é uma pilha TCP/IP de código aberto desenvolvida especificamente para dispositivos embarcados. O foco de sua implementação é reduzir o consumo de memória RAM, mantendo as funcionalidades de uma pilha TCP/IP completa (STMicroelectronics, 2016b).

O LwIP oferece diferentes interfaces de programação (APIs) para a aplicação. Para arquiteturas *bare-metal*, a *Raw API* (API nativa) permite o desenvolvimento de aplicações baseadas em *callbacks* de eventos. Essa API fornece o melhor desempenho e tamanho de código otimizado, garantindo um processamento determinístico (STMicroelectronics, 2016b).

Na construção de interfaces de monitoramento, como servidores HTTP embarcados, a tecnologia *Server Side Includes* (SSI) é um método utilizado para incluir dados

dinâmicos no código HTML da página. O processo funciona mediante a alocação de uma *tag* (marcador) específica dentro do código HTML. Quando o servidor recebe uma requisição para a página da web, ele a analisa e, ao encontrar a respectiva *tag*, a substitui dinamicamente pelo valor calculado no microcontrolador antes de enviar a resposta ao cliente (STMicroelectronics, 2016b).

2.3.3 Ambiente de Desenvolvimento STM32CubeIDE

O STM32CubeIDE é um Ambiente de Desenvolvimento Integrado (IDE, *Integrated Development Environment*) desenvolvido pela STMicroelectronics para programação e depuração de microcontroladores STM32 (STMicroelectronics, 2021). A ferramenta integra edição de código, compilação e depuração em um único ambiente, com suporte às bibliotecas de abstração de hardware (HAL, *Hardware Abstraction Layer*).

O ecossistema STM32Cube inclui o STM32CubeMX, interface gráfica que permite configurar periféricos, clock e middlewares¹ do dispositivo, gerando automaticamente o código inicial do projeto.

2.4 TRABALHOS RELACIONADOS

A detecção rápida de afundamentos de tensão tem sido objeto de investigação contínua na literatura, com diferentes abordagens propostas para equilibrar velocidade de resposta, robustez e viabilidade de implementação.

O artigo (Naidoo; Pillay, 2007) aborda um método adaptativo não linear capaz de estimar simultaneamente amplitude, fase e frequência do sinal de tensão em tempo real, com latências reportadas entre 1 ms e 4 ms para diferentes magnitudes e ângulos de incidência da falta. Os autores realizaram comparações experimentais com métodos RMS de um ciclo e tensão de pico, estabelecendo uma referência quantitativa amplamente utilizada na área. Contudo, o algoritmo proposto exige o rastreamento contínuo de variáveis de estado por equações diferenciais discretizadas, impondo custo computacional relevante para arquiteturas embarcadas sem sistema operacional.

No trabalho (Sillapawicharn, 2015) é proposto um detector ortogonal baseado em diferenciador, que sintetiza uma onda cosseno a partir da tensão de entrada para calcular a amplitude instantânea. O método alcança tempo médio de detecção de 0,33 ms para afundamentos de 0,5 pu, com pior caso de 0,88 ms. Apesar da elevada velocidade, o diferenciador amplifica componentes de alta frequência, tornando o método suscetível a ruídos e harmônicos presentes no sinal, limitação apontada pelo próprio autor.

¹ *Middlewares* são camadas de software intermediárias que abstraem funcionalidades complexas do hardware, oferecendo interfaces padronizadas para o desenvolvimento de aplicações. No contexto deste trabalho, destaca-se o uso do LwIP (*Lightweight IP*), um middleware de pilha TCP/IP de código aberto otimizado para sistemas embarcados com recursos limitados de memória, responsável por viabilizar a comunicação via protocolo Ethernet no dispositivo STM32.

O estudo (Guan *et al.*, 2024) foca na detecção de afundamentos assimétricos em redes trifásicas com componentes de sequência negativa e harmônicos de ordem ímpar, propondo a combinação do método de atraso de pequeno ângulo com filtro de média móvel de meio ciclo. A abordagem obtém separação de sequências com atraso de 1 ms, acrescido de 10 ms pela etapa de filtragem harmônica, demonstrando que o tratamento de cenários assimétricos exige estágios adicionais de processamento com impacto direto na latência total.

Em síntese, observa-se que a literatura apresenta métodos capazes de alcançar tempos de detecção inferiores a 1 ms, porém frequentemente à custa de maior complexidade algorítmica, maior sensibilidade a ruídos ou necessidade de estágios adicionais de processamento. Nesse contexto, o presente trabalho concentra-se na avaliação comparativa de cinco algoritmos de detecção sob diferentes condições de afundamento, priorizando soluções compatíveis com implementação embarcada de baixo custo computacional. Além disso, o método selecionado foi implementado em um microcontrolador STM32F767 operando em *bare-metal*², integrado a uma interface web embarcada para monitoramento remoto dos eventos detectados. A Tabela 1 resume as principais características dos trabalhos analisados e do método proposto, destacando critérios como complexidade computacional, tempo de resposta, validação experimental, análise para diferentes ângulos de incidência da falta (POW) e avaliação da influência da magnitude nos resultados obtidos.

Tabela 1 – Comparativo entre trabalhos relacionados e o presente trabalho

Trabalho	Compl. \mathcal{O}	Tempo de Res- posta baixo	Valid. experi- mental	Análise POW	Infl. magni- tude
(Naidoo; Pillay, 2007)	Alta	✗	✓	✗	✗
(Sillapawicharn, 2015)	Baixa	✓	✗	✓	✓
(Guan <i>et al.</i> , 2024)	Média	✗	✗	✓	✗
Este trabalho	$\mathcal{O}(1)$	✓	✓	✓	✓

Fonte: Elaborado pelo autor.

² Programação *bare-metal* refere-se à execução do código diretamente sobre o hardware, sem a intermediação de um sistema operacional. Dessa forma, o software interage diretamente com os registradores e periféricos do microcontrolador, o que proporciona controle total sobre o tempo de execução e elimina as latências introduzidas por camadas de abstração de sistema operacional, característica essencial para aplicações com requisitos de tempo real determinístico.

3 METODOLOGIA

A metodologia adotada neste trabalho possui caráter experimental e aplicado, com o objetivo de avaliar diferentes métodos matemáticos para detecção de afundamentos de tensão (*voltage sag*) em sistemas de monitoramento de qualidade de energia. O estudo envolve duas etapas principais: (i) avaliação comparativa de algoritmos de detecção em ambiente de simulação e (ii) implementação prática do método selecionado em um sistema embarcado baseado em microcontrolador.

O desenvolvimento foi conduzido de forma sequencial, permitindo validar inicialmente os algoritmos em ambiente computacional controlado e posteriormente aplicar os mesmos conceitos em uma implementação embarcada para operação em tempo real.

3.1 VISÃO GERAL DA METODOLOGIA

A metodologia do trabalho foi estruturada em três etapas principais:

1. **Modelagem e simulação:** implementação dos algoritmos em linguagem C para avaliação comparativa do tempo de detecção de afundamentos de tensão.
2. **Implementação embarcada:** desenvolvimento de um sistema de aquisição e processamento utilizando um microcontrolador STM32.
3. **Monitoramento e visualização:** disponibilização das informações detectadas por meio de uma interface web embarcada.

3.2 MÉTODOS PARA DETECÇÃO DE AFUNDAMENTOS DE TENSÃO

Neste trabalho, foram avaliados cinco métodos, abrangendo o cálculo do valor eficaz (RMS) em janelas fixas e móveis e a análise dos valores de pico da forma de onda em janelas de ciclo completo e meio ciclo, conforme detalhado nas subseções a seguir.

3.2.1 Método RMS em Bloco

O valor RMS (*Root Mean Square*) é amplamente utilizado para representar a magnitude eficaz de sinais de tensão e corrente em sistemas elétricos (Tunaboylu; Collins; Chaney, 1998). No método RMS em bloco, o valor RMS é calculado utilizando uma janela fixa correspondente a um ciclo completo da frequência fundamental.

A tensão RMS é calculada por:

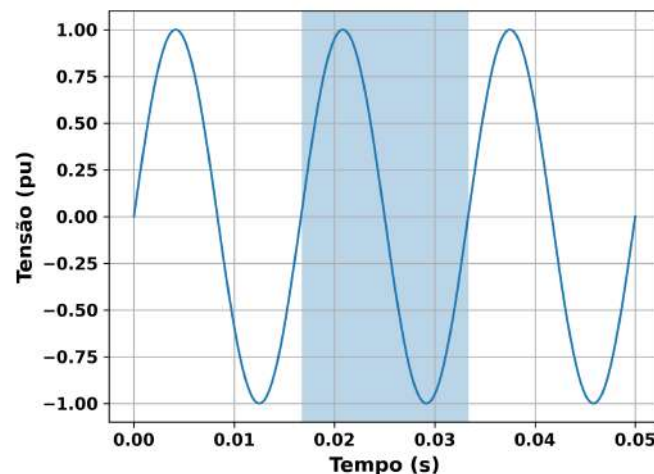
$$V_i^{rms} = \sqrt{\frac{1}{N} \sum_{j=i}^{i+N-1} V_j^2}, \quad (3.1)$$

onde N representa o número de amostras por ciclo da frequência fundamental, V_j é o valor da tensão amostrada no instante j , e V_i^{rms} é o valor RMS calculado para a janela iniciada na amostra i .

Nesse método, um ciclo completo da forma de onda de tensão é coletado e processado para obtenção do valor RMS. A detecção do afundamento de tensão é realizada comparando o valor RMS calculado pela Equação (3.1) com o limiar definido na Seção 2.1.1. Como esse método utiliza uma janela de um ciclo completo, pode ocorrer um atraso de até um ciclo entre o início real da perturbação e sua detecção (Naidoo; Pillay, 2007).

A Figura 2 ilustra a janela de cálculo utilizada nesse método. A região sombreada representa o intervalo correspondente a um ciclo completo da forma de onda considerado no cálculo do valor RMS.

Figura 2 – Janela de cálculo do método RMS em bloco utilizando um ciclo completo da forma de onda.



Fonte: Elaborado pelo autor.

3.2.2 Tensão de Pico em um Ciclo Completo

Outra abordagem para detecção de afundamentos de tensão baseia-se no valor de pico da forma de onda de tensão (Naidoo; Pillay, 2007; Sillapawicharn, 2015). Nesse método, determina-se o maior valor absoluto da tensão dentro de uma janela correspondente a um ciclo completo.

O valor máximo da tensão é obtido conforme descrito pela Equação (3.2), que define o maior valor absoluto da forma de onda dentro do intervalo analisado.

$$V_{peak} = \max(|V(t)|) \quad (3.2)$$

Ao monitorar o valor máximo da tensão dentro de cada ciclo, reduções significativas nesse valor podem indicar a ocorrência de um afundamento de tensão. Esse método apresenta implementação simples e baixo custo computacional.

3.2.3 Tensão de Pico em Meio Ciclo

Uma variação do método de tensão de pico consiste em avaliar o valor máximo da tensão dentro de uma janela correspondente a meio ciclo da frequência fundamental (Naidoo; Pillay, 2007). Essa abordagem permite uma detecção mais rápida das perturbações em comparação com o método baseado em ciclo completo.

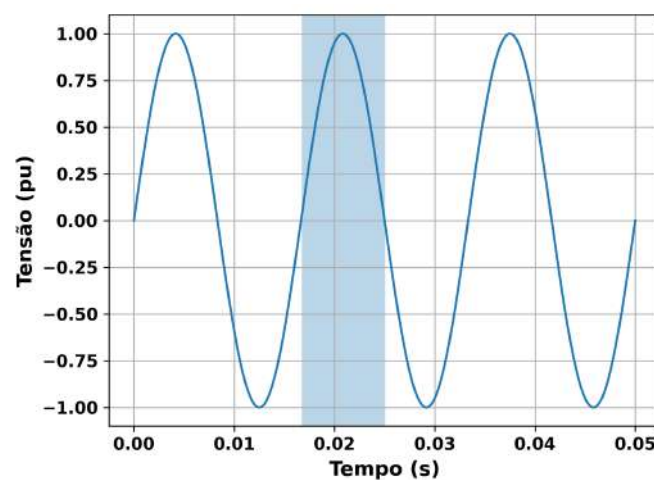
A tensão de pico pode ser expressa por:

$$V_{peak} = \max |V(t - \tau)|, \quad 0 < \tau < \frac{T}{2}, \quad (3.3)$$

onde na Equação (3.3), T representa o período da frequência fundamental e τ corresponde ao intervalo de tempo dentro do meio ciclo anterior. Como a janela de análise é menor, esse método apresenta maior rapidez na identificação de variações abruptas na tensão.

A Figura 3 apresenta a janela de análise correspondente a meio ciclo utilizada nesse método.

Figura 3 – Janela de análise utilizada no método de tensão de pico em meio ciclo.



Fonte: Elaborado pelo autor.

3.2.4 Método RMS Móvel com Janela de Um Ciclo

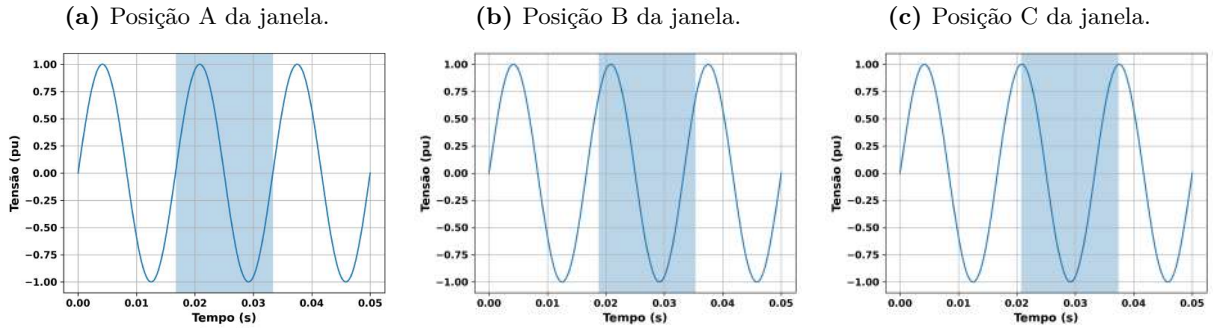
Além do cálculo RMS em bloco, foi implementado um método baseado em RMS móvel. Nesse caso, utiliza-se uma janela deslizante contendo um ciclo completo de amostras da tensão.

A cada nova amostra adquirida, o valor mais antigo da janela é removido e o novo valor é inserido, formando um vetor de comprimento N . O valor RMS é então recalculado utilizando a mesma relação matemática apresentada na Equação (3.1).

Esse método equivale a uma média móvel dos valores quadráticos da tensão ao longo de um ciclo completo. O uso de uma janela deslizante permite maior resolução temporal na detecção de afundamentos de tensão em comparação com o método RMS em bloco.

A Figura 4 ilustra o deslocamento da janela de cálculo ao longo do tempo.

Figura 4 – Deslocamento da janela de cálculo no método RMS móvel.



Fonte: Elaborado pelo autor.

Nas três subfiguras apresentadas observa-se o deslocamento progressivo da janela de cálculo. Na posição (a) a janela contém um determinado conjunto de amostras da forma de onda. Na posição (b) a janela é deslocada após a aquisição de uma nova amostra, removendo o valor mais antigo e incorporando o novo valor adquirido. O mesmo processo ocorre na posição (c), mantendo sempre o mesmo número de amostras dentro da janela.

Esse mecanismo permite atualizar continuamente o valor RMS estimado, proporcionando maior resolução temporal na detecção de afundamentos de tensão.

3.2.5 Método RMS Móvel com Janela de Meio Ciclo

Também foi implementada uma variação do método RMS móvel utilizando uma janela correspondente a meio ciclo da forma de onda. Nesse caso, o vetor utilizado no cálculo contém apenas metade das amostras de um ciclo.

O valor RMS é então calculado por:

$$V_i^{rms} = \sqrt{\frac{1}{M} \sum_{j=i}^{i+M-1} V_j^2}, \quad (3.4)$$

onde na Equação (3.4), $M = \frac{N}{2}$ representa o número de amostras correspondente a meio ciclo.

A redução do tamanho da janela permite diminuir o tempo de detecção do afundamento de tensão. Entretanto, janelas menores podem tornar o método mais sensível a distorções da forma de onda e ruídos presentes no sinal (Guan *et al.*, 2024).

3.3 SIMULAÇÃO E AVALIAÇÃO DOS ALGORITMOS

Para avaliar o desempenho dos métodos propostos na Seção 3.2 e fundamentar a escolha do algoritmo a ser embarcado, desenvolveu-se um programa computacional

em linguagem C. Este ambiente de simulação foi responsável por emular a aquisição digital do sinal de tensão e processá-lo simultaneamente com as cinco técnicas de detecção abordadas.

Para validar a capacidade de resposta dos algoritmos sob diferentes cenários operacionais, inseriram-se eventos transitórios de afundamento de tensão no sinal simulado. As características dessas perturbações, como a severidade da redução de amplitude e os instantes de início e fim do evento, foram parametrizadas no código, permitindo a execução de múltiplos testes independentes.

3.3.1 Critério e registros adotados

Com base na definição apresentada na Seção 2.1.1, este trabalho adotou o limiar de 0,9 pu como valor de referência para identificação do início do evento de afundamento. A detecção é confirmada quando a magnitude estimada pela técnica avaliada cai abaixo desse limiar, e o término do evento é registrado no instante em que a tensão retorna acima do mesmo valor.

Durante a ocorrência do evento, o sistema registra:

- instante de início do afundamento;
- instante de recuperação da tensão;
- duração total do evento.

Ao processar o sinal, o sistema registrou em um arquivo de dados o exato instante temporal em que cada algoritmo sinalizou a anomalia. O objetivo central dessa abordagem foi extrair e comparar o tempo de latência de cada método, definido como a diferença entre o início real do afundamento e o instante de sua detecção, permitindo identificar qual técnica oferece a resposta mais rápida para implementação em tempo real (Sillapawicharn, 2015; Naidoo; Pillay, 2007).

3.4 SISTEMA DE AQUISIÇÃO DO SINAL

Para possibilitar a detecção de afundamentos de tensão, tornou-se necessário adquirir o sinal elétrico da rede e convertê-lo para o formato digital. Esse processo é uma etapa fundamental, pois permite que técnicas de processamento digital de sinais sejam aplicadas posteriormente na análise do comportamento da tensão.

A aquisição foi realizada por meio do conversor analógico-digital (ADC) integrado ao microcontrolador STM32F767. Esse dispositivo foi escolhido devido à sua elevada capacidade de processamento e à disponibilidade de periféricos adequados para aplicações em tempo real.

3.4.1 Frequência do sinal monitorado

Tanto o sistema quanto a simulação foram projetados considerando uma rede elétrica com frequência fundamental de $f = 60$ Hz, valor correspondente ao padrão adotado em sistemas elétricos de potência em diversos países.

3.4.2 Taxa de amostragem

Para garantir representação adequada da forma de onda da tensão, foi utilizada uma taxa de amostragem significativamente superior à frequência do sinal.

A aquisição foi realizada com frequência de amostragem de $f_s = 10$ kHz, configurando um temporizador interno do microcontrolador para acionar o ADC a cada $T_s = 100 \mu s$.

Com essa taxa de amostragem, obtém-se aproximadamente o número de amostras por ciclo da rede elétrica, conforme definido pela Equação (3.5).

$$N = \frac{f_s}{f} \approx \frac{10000}{60} \approx 167 \quad (3.5)$$

Esse resultado fornece resolução temporal adequada para a estimativa da magnitude da tensão e para a detecção de variações abruptas no sinal. Para a implementação computacional, adotou-se uma janela fixa de $N = 167$ amostras por ciclo. Embora a divisão resulte em um valor não inteiro, a adoção de $N = 167$ amostras implica um erro relativo inferior a 0,5%, considerado desprezível para aplicações de detecção de afundamentos de tensão, não comprometendo a precisão da estimativa RMS.

3.4.3 Geração do sinal de teste

Para avaliar o funcionamento do sistema embarcado em condições controladas, foi utilizado um gerador de sinais Agilent 33521A para produzir a forma de onda aplicada à entrada do conversor analógico-digital do microcontrolador.

O gerador foi configurado para produzir um sinal senoidal com frequência de 60 Hz, amplitude de 3 V pico a pico (V_{pp}) e tensão de offset de 1,65 V. Essa configuração foi escolhida para adequar o sinal à faixa de operação do ADC do microcontrolador, garantindo que toda a forma de onda permanecesse dentro da região de tensão positiva.

Dessa forma, o sinal aplicado ao microcontrolador apresenta uma forma de onda senoidal deslocada, variando aproximadamente entre 0 V e 3,3 V, compatível com o intervalo de entrada do ADC do microcontrolador. Esse tipo de condicionamento também é compatível com módulos comerciais de medição de tensão, como o sensor ZMPT101B.

O sinal gerado foi conectado diretamente à entrada analógica do microcontrolador, permitindo que o sistema realizasse a aquisição contínua das amostras e executasse os algoritmos de detecção de afundamento de tensão descritos nas seções seguintes.

3.4.4 Aquisição com temporizador e DMA

A aquisição determinística de sinais foi implementada por meio da operação coordenada de três periféricos principais, formando um pipeline eficiente de captura de dados (STM32F76XXX... , 2024): o temporizador de hardware, o conversor analógico-digital (ADC) e o controlador de acesso direto à memória (DMA).

Conforme fundamentado anteriormente, a amostragem foi gerenciada por um temporizador de hardware. Especificamente, o periférico foi alimentado por um clock de 108 MHz e parametrizado com um *prescaler* nulo, operando em modo de contagem crescente e com o período definido em 10799 (totalizando 10800 ciclos da base de tempo). Essa configuração permite gerar eventos periódicos de acionamento do ADC, garantindo o determinismo temporal necessário para o cálculo preciso do valor eficaz (RMS) sem depender do fluxo de execução do software principal.

O ADC, por sua vez, realiza a conversão do sinal analógico contínuo em uma sequência de amostras discretas. A taxa de amostragem foi definida de modo a capturar adequadamente a forma de onda da rede elétrica, permitindo a identificação de distúrbios transitórios de curta duração, como os afundamentos de tensão.

Para complementar o processo, os dados convertidos pelo ADC foram transferidos para a memória RAM por meio do DMA. Conforme fundamentado, essa abordagem elimina a intervenção da CPU na aquisição.

A integração entre temporizador, ADC e DMA garante uma aquisição contínua, eficiente e determinística, sendo uma abordagem amplamente empregada em sistemas de processamento digital de sinais em tempo real. Dentre suas principais vantagens, destacam-se:

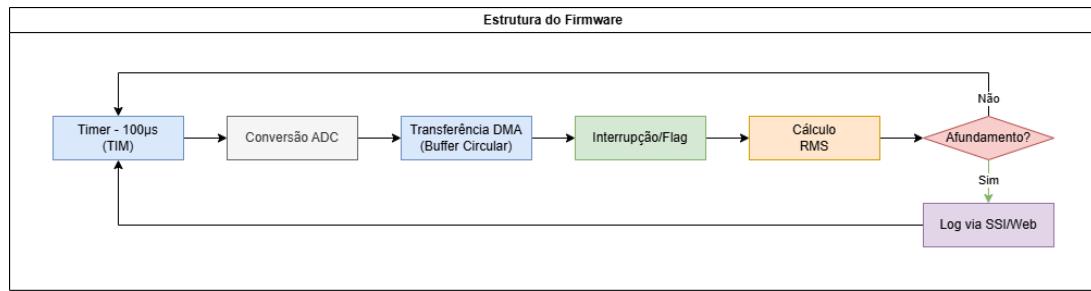
- redução da carga de processamento da CPU;
- aquisição contínua e determinística do sinal.

3.5 IMPLEMENTAÇÃO EMBARCADA

Finalizada a etapa de simulação e avaliação dos algoritmos, procedeu-se à implementação prática do sistema utilizando o microcontrolador STM32F767ZI. O firmware foi desenvolvido inteiramente em linguagem C¹, sendo estruturado para executar quatro funções fundamentais de forma cíclica e determinística: aquisição, processamento, detecção e registro de eventos.

A arquitetura geral e o fluxo de dados do sistema embarcado são apresentados no diagrama da Figura 5.

¹ O código-fonte completo do *firmware* desenvolvido neste trabalho está disponível publicamente em: https://github.com/LucasMedeirosC/TCC_STM32.

Figura 5 – Diagrama de blocos funcional da implementação embarcada.

Fonte: Elaborado pelo autor.

A implementação foi realizada no ambiente de desenvolvimento STM32CubeIDE (versão 1.19.0), no qual a lógica principal do processamento foi executada em rotinas acionadas por interrupções associadas à aquisição de dados, garantindo determinismo temporal.

Em termos de funcionamento, o sistema inicia com a aquisição contínua do sinal de tensão da rede elétrica, o qual é convertido para o formato digital por meio do ADC integrado ao microcontrolador. Em seguida, as amostras são processadas utilizando o algoritmo de detecção previamente selecionado na etapa de simulação, permitindo o monitoramento da magnitude da tensão.

Com base nesse processamento, o sistema realiza a detecção de eventos de afundamento sempre que os critérios estabelecidos são atendidos. Por fim, ao identificar um evento, são registradas informações relevantes, como data, hora e duração, possibilitando posterior análise do distúrbio.

3.6 INTERFACE WEB PARA MONITORAMENTO

Para possibilitar que o usuário acesse e visualize os dados sobre os eventos de afundamento registrados pelo firmware descrito na seção anterior, foi implementada uma interface web embarcada no sistema.

A interface foi desenvolvida utilizando páginas HTML com suporte a *Server Side Includes* (SSI), permitindo que o microcontrolador insira dinamicamente informações atualizadas nas páginas exibidas ao usuário.

A interface apresenta as seguintes informações:

- data e hora de ocorrência do afundamento;
- tipo da ocorrência do afundamento;
- duração do evento em segundos.

Essa abordagem permite o monitoramento remoto dos eventos registrados pelo sistema utilizando apenas um navegador web conectado à rede.

3.7 SÍNTESE DA METODOLOGIA

A metodologia foi estruturada em três etapas sequenciais. Na primeira, os algoritmos foram implementados em ambiente de simulação computacional, permitindo comparar o tempo de detecção de cada método sob condições controladas. Na segunda, o algoritmo de melhor desempenho foi transposto para o microcontrolador STM32F767, onde a aquisição e o processamento do sinal foram validados em tempo real. Na terceira, os eventos detectados foram disponibilizados por meio de uma interface web embarcada. Os resultados obtidos em cada etapa são apresentados no capítulo a seguir.

4 RESULTADOS

Conforme apresentado no capítulo anterior, foram avaliados diferentes métodos matemáticos para detecção de afundamentos de tensão (*voltage sag*). Inicialmente, os algoritmos foram implementados em linguagem C e executados em ambiente de simulação, permitindo comparar o tempo de latência de cada método para identificar a ocorrência da anomalia.

Após a etapa de validação computacional, o algoritmo com melhor desempenho foi embarcado em um microcontrolador STM32F767 para operação em tempo real, em conjunto com uma interface web para monitoramento remoto dos eventos.

4.1 AVALIAÇÃO DOS MÉTODOS DE DETECÇÃO EM SIMULAÇÃO

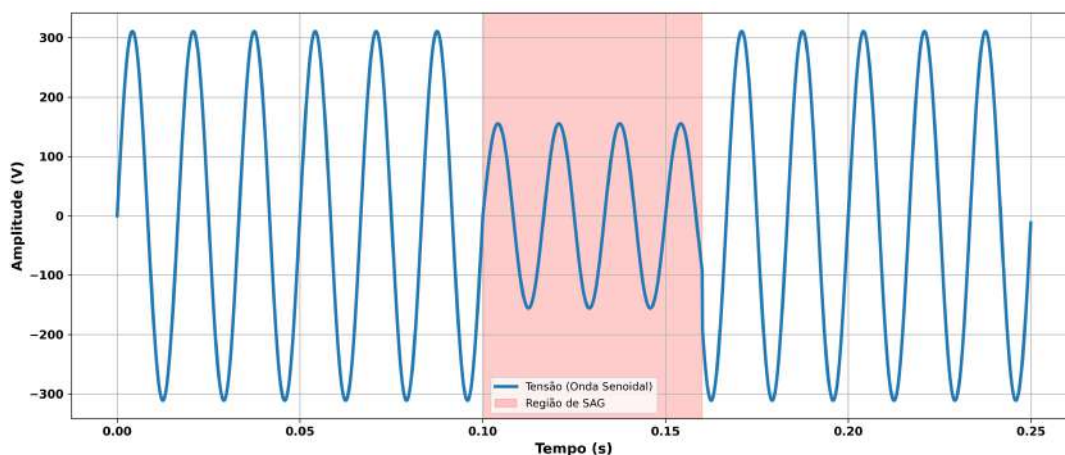
Para uma avaliação abrangente, os métodos de detecção foram submetidos a diferentes condições operacionais. Como o tempo de detecção de alguns algoritmos é fortemente dependente do ângulo de fase da tensão no instante da falta, a análise inicial foi dividida em três cenários distintos de afundamento, variando o ponto de início do evento na forma de onda:

- **Cenário 1:** Afundamento com início em 0° (cruzamento por zero).
- **Cenário 2:** Afundamento com início em 45° (rampa de subida).
- **Cenário 3:** Afundamento com início em 90° (pico positivo).

Para estes três cenários, o evento foi simulado reduzindo a amplitude do sinal em 50%, adotando-se o limiar de detecção de $V < 0,9$ pu. Em seguida, avaliou-se também a influência da magnitude do afundamento sobre o tempo de detecção.

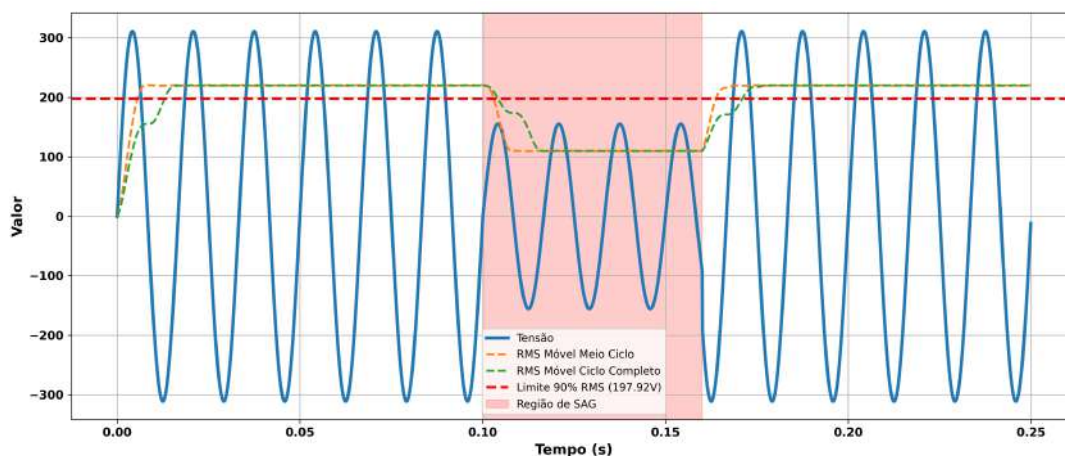
4.1.1 Cenário 1: Início do afundamento em 0 graus

Neste primeiro cenário, a perturbação foi inserida exatamente no momento em que a forma de onda de tensão cruza o eixo zero com derivada positiva (ângulo de 0°), o que corresponde ao instante de 0,1001 s na simulação. A Figura 6 ilustra a forma de onda simulada, destacando a região do afundamento.

Figura 6 – Sinal de tensão com inserção de afundamento em 0° (cruzamento por zero).

Fonte: Elaborado pelo autor.

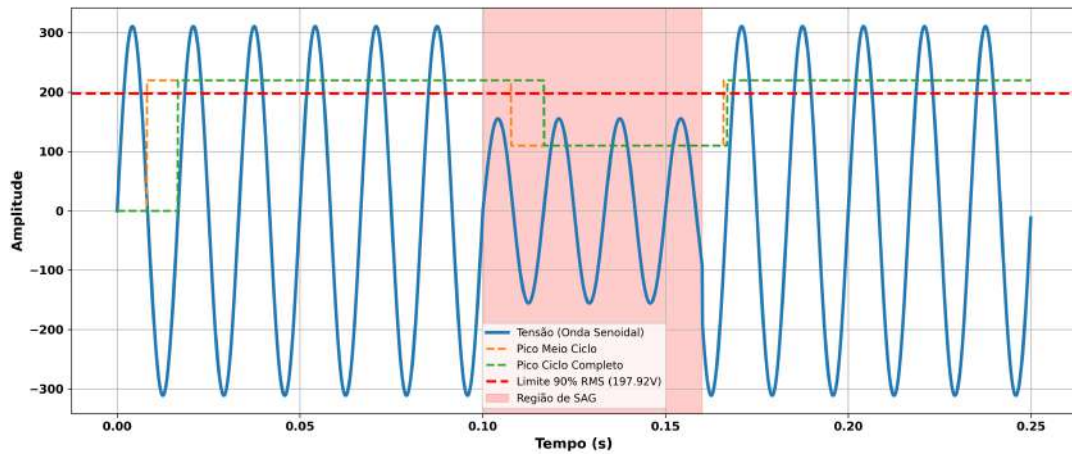
Ao aplicar os algoritmos de detecção sobre este sinal, observam-se comportamentos distintos dependendo da natureza matemática do método. A Figura 7 apresenta a resposta dos métodos baseados em RMS Móvel. Nota-se que, como a janela desliza a cada nova amostra, o valor RMS estimado começa a decair logo após o início da anomalia, cruzando rapidamente o limiar de 0,9 pu.

Figura 7 – Comportamento da detecção utilizando janelas de RMS Móvel (Cenário 0°).

Fonte: Elaborado pelo autor.

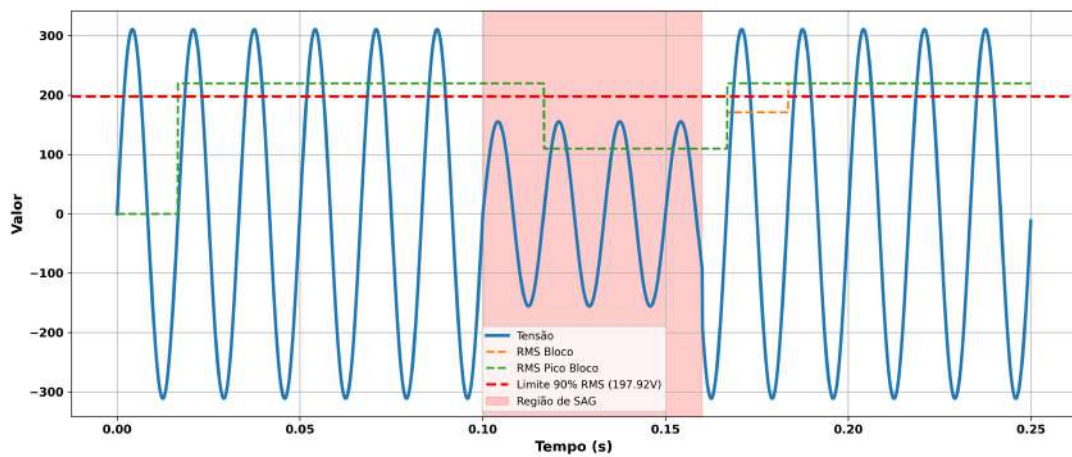
Por outro lado, os métodos baseados em janelas fixas (Bloco) apresentam uma limitação temporal severa e síncrona neste cenário. Como a falha ocorre no cruzamento por zero, os algoritmos baseados em bloco completo (RMS em Bloco e Pico em Bloco de ciclo completo) necessitam aguardar a finalização integral do intervalo de integração para atualizar seus valores de saída. Devido a essa restrição de fronteira, ambos os métodos de ciclo completo registram exatamente o mesmo tempo de atraso de 16,7 ms. De forma análoga, o método de meio ciclo fixo fica limitado ao término da sua respectiva metade de janela. A Figura 8 ilustra o comportamento dos métodos de pico, enquanto a Figura 9 demonstra a evolução estática do cálculo baseado em Blocos.

Figura 8 – Comportamento da detecção utilizando métodos de Tensão de Pico (Cenário 0°).



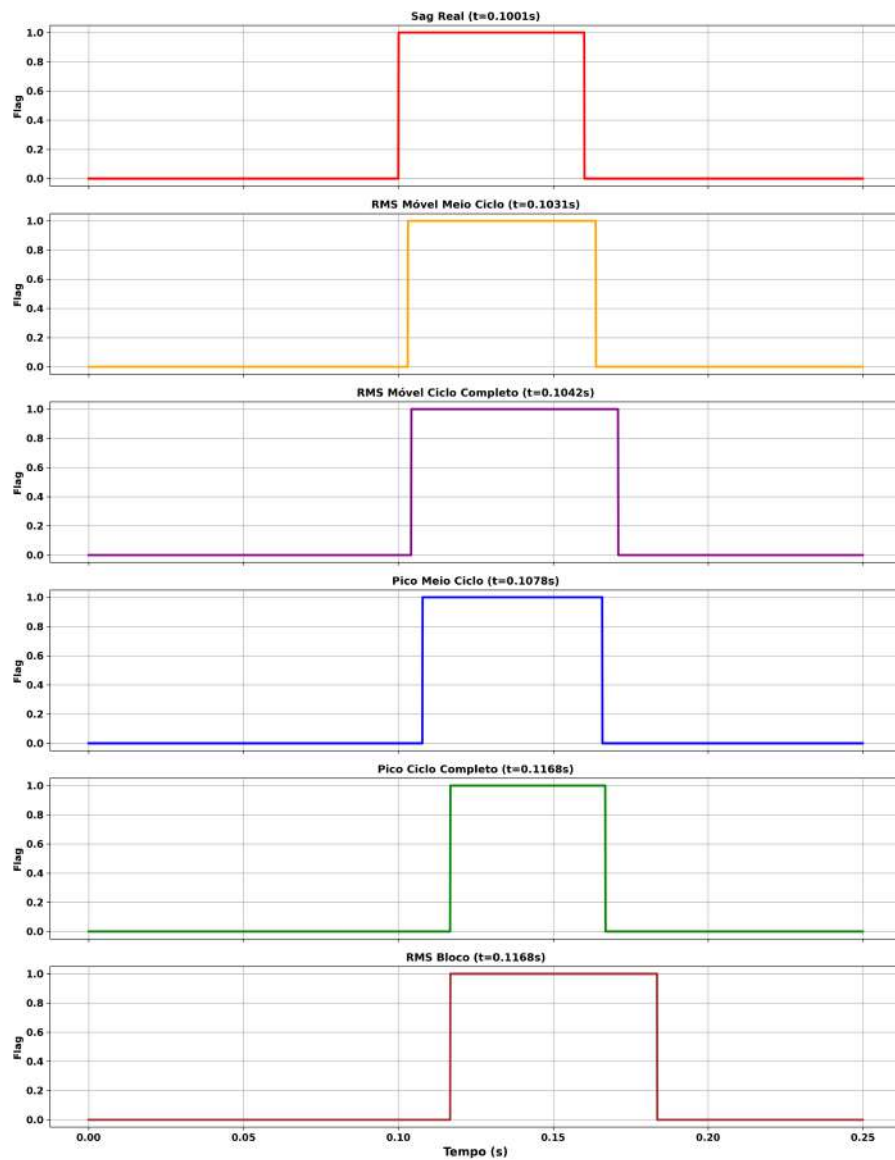
Fonte: Elaborado pelo autor.

Figura 9 – Comportamento da detecção utilizando cálculo RMS em Bloco vs Pico em Bloco (Cenário 0°).



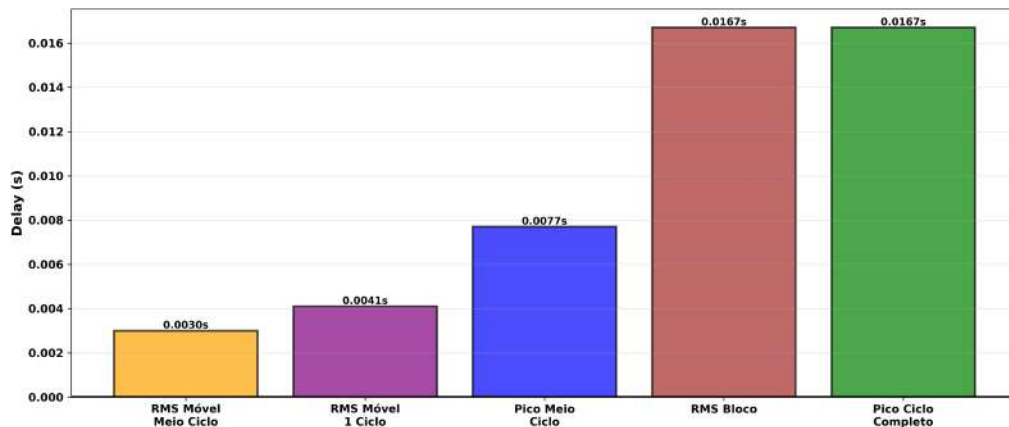
Fonte: Elaborado pelo autor.

O disparo das *flags* de detecção de todos os métodos para este cenário está consolidado na Figura 10.

Figura 10 – Sinalização (*flags*) de detecção dos algoritmos (Cenário 0°).

Fonte: Elaborado pelo autor.

A quantificação do erro (atraso de detecção) em relação ao instante real da falha revela o menor tempo de resposta dos métodos móveis para faltas em cruzamento por zero. A Figura 11 e a Tabela 2 detalham os atrasos calculados.

Figura 11 – Atraso de detecção para cada algoritmo no Cenário 1 (0°).

Fonte: Elaborado pelo autor.

Tabela 2 – Resumo do atraso de detecção no Cenário 1 (0°).

Método	Atraso de detecção (ms)
RMS Móvel (Meio Ciclo)	3,0
RMS Móvel (1 Ciclo)	4,1
Pico (Meio Ciclo)	7,7
RMS Bloco	16,7
Pico (1 Ciclo Completo)	16,7

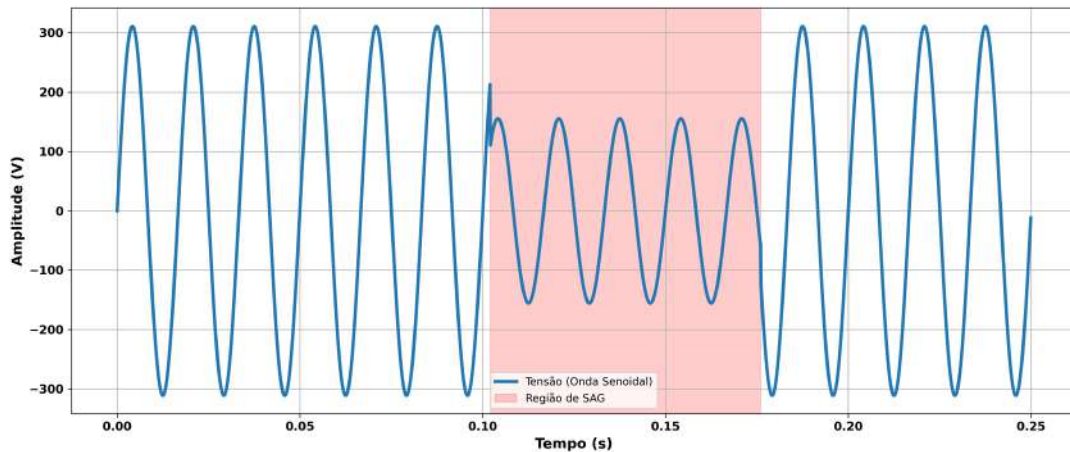
Fonte: Elaborado pelo autor.

Neste cenário crítico para detecção (cruzamento por zero), o algoritmo RMS Móvel de Meio Ciclo apresentou a resposta mais ágil, sinalizando a perturbação em apenas 3,0 ms. Em contrapartida, os métodos baseados em processamento de um ciclo completo (RMS em Bloco e Pico de Ciclo Completo) apresentaram o pior desempenho, empatados com 16,7 ms de atraso, o que corresponde a exatamente um ciclo completo da frequência fundamental de 60 Hz.

4.1.2 Cenário 2: Início do afundamento em 45 graus

O segundo cenário de simulação avaliou o comportamento dos algoritmos quando a perturbação tem início na rampa de subida da forma de onda, correspondendo a um ângulo elétrico de 45° . O evento foi inserido no instante $t = 0,1021$ s, conforme ilustrado na Figura 12. Este cenário atua como um ponto intermediário e crítico entre a ausência de sinal no zero e a energia máxima no pico.

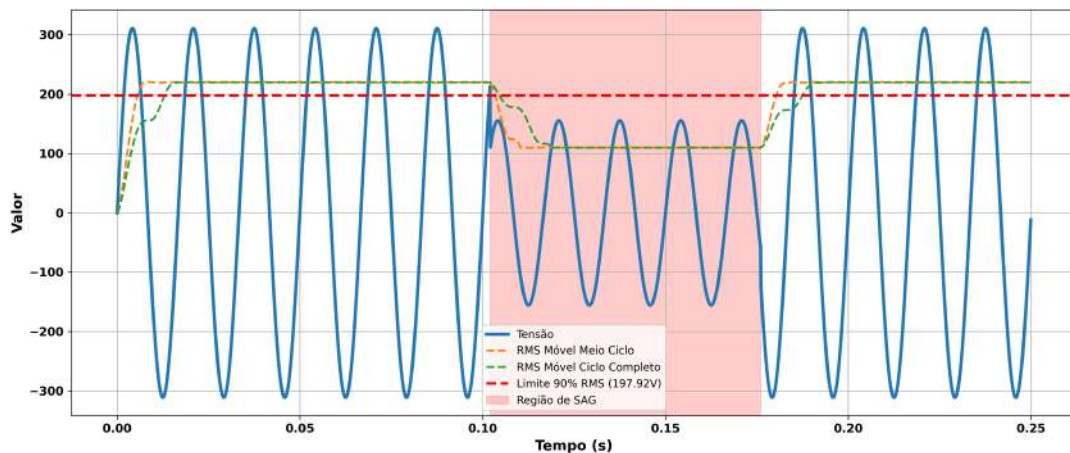
Figura 12 – Sinal de tensão com inserção de afundamento em 45°.



Fonte: Elaborado pelo autor.

Por ocorrer em uma região de crescimento da onda, a queda de tensão reflete-se com moderada rapidez nas janelas de integração. A Figura 13 detalha a transição do valor calculado para as janelas deslizantes de RMS, demonstrando uma resposta ágil.

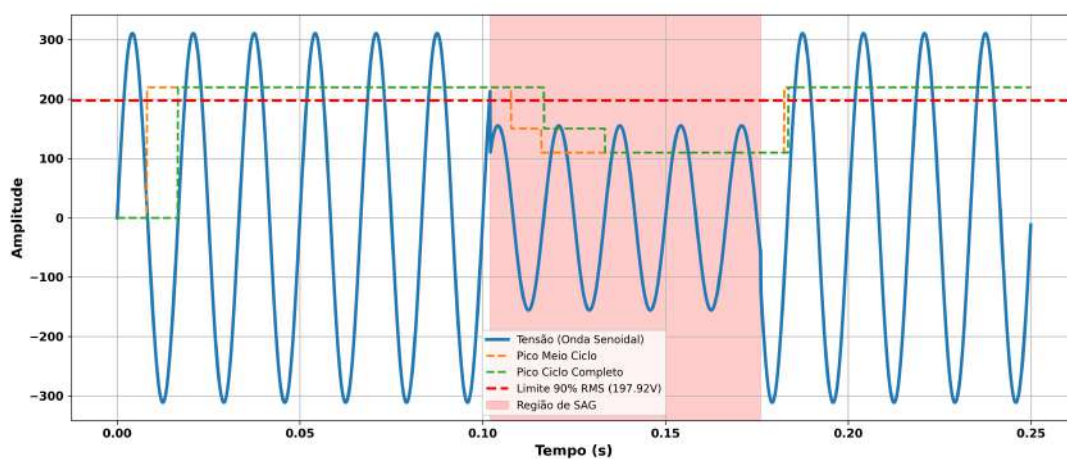
Figura 13 – Comportamento da detecção utilizando janelas de RMS Móvel (Cenário 45°).



Fonte: Elaborado pelo autor.

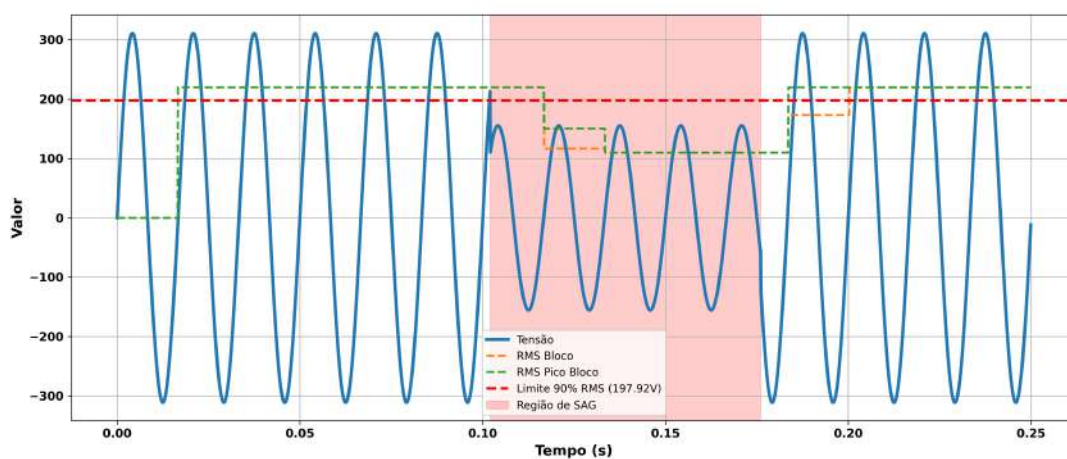
A dinâmica dos algoritmos que dependem do preenchimento de janelas fixas e análise de picos é apresentada nas Figuras 14 e 15. As respostas lógicas do sistema (*flags*) podem ser visualizadas na Figura 16.

Figura 14 – Comportamento da detecção utilizando métodos de Tensão de Pico (Cenário 45°).

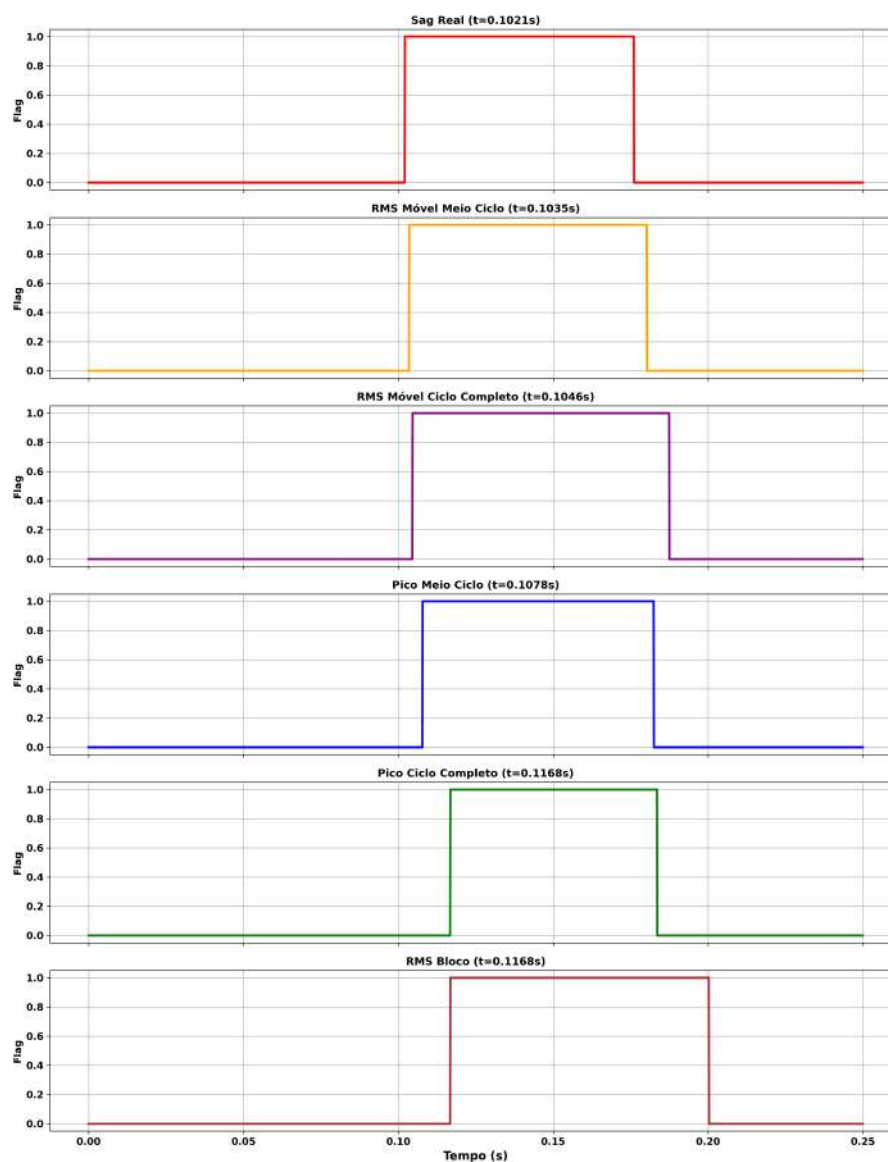


Fonte: Elaborado pelo autor.

Figura 15 – Comportamento da detecção utilizando cálculo RMS em Bloco vs Pico em Bloco (Cenário 45°).

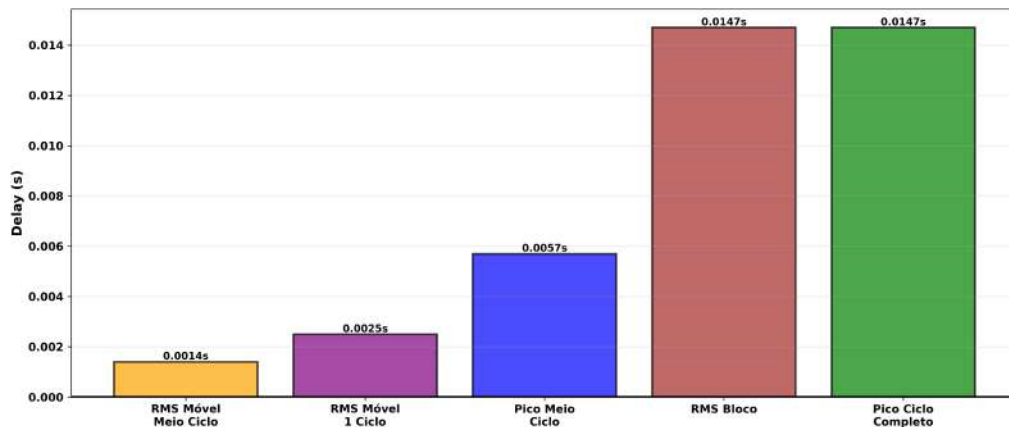


Fonte: Elaborado pelo autor.

Figura 16 – Sinalização (*flags*) de detecção dos algoritmos (Cenário 45°).

Fonte: Elaborado pelo autor.

A quantificação dos atrasos de detecção para este cenário é apresentada na Figura 17 e na Tabela 3.

Figura 17 – Atraso de detecção para cada algoritmo no Cenário 2 (45°).

Fonte: Elaborado pelo autor.

Tabela 3 – Resumo do atraso de detecção no Cenário 2 (45°).

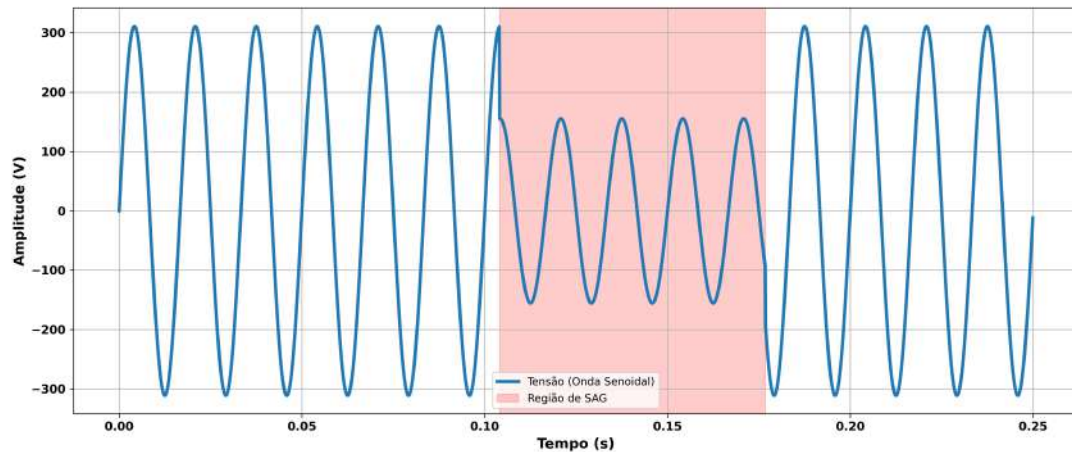
Método	Atraso de detecção (ms)
RMS Móvel (Meio Ciclo)	1,4
RMS Móvel (1 Ciclo)	2,5
Pico (Meio Ciclo)	5,7
RMS Bloco	14,7
Pico (1 Ciclo Completo)	14,7

Fonte: Elaborado pelo autor.

Neste cenário, o RMS Móvel de Meio Ciclo confirmou melhor desempenho com um tempo de resposta de apenas 1,4 ms. Os métodos processados em janelas fixas completas de um ciclo acumularam atrasos de 14,7 ms.

4.1.3 Cenário 3: Início do afundamento em 90 graus

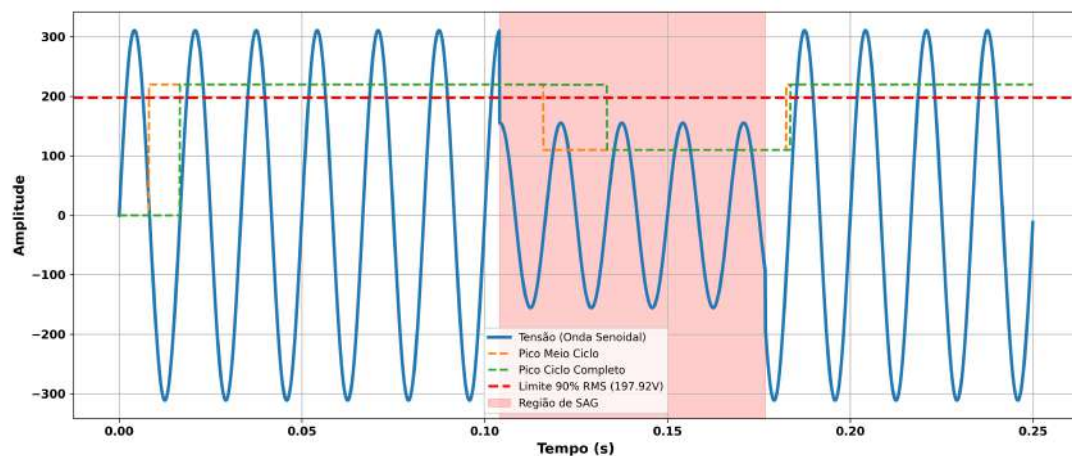
No terceiro cenário avaliado, o afundamento de tensão foi inserido no exato instante em que a forma de onda atinge o seu valor de pico positivo, correspondendo a um ângulo elétrico de 90°. O evento ocorreu no instante $t = 0,1041$ s na simulação. A Figura 18 ilustra a característica do sinal para esta condição, onde a queda de amplitude ocorre em um ponto de máxima energia.

Figura 18 – Sinal de tensão com inserção de afundamento em 90° (pico positivo).

Fonte: Elaborado pelo autor.

O comportamento dos algoritmos sob esta nova condição apresenta o pior caso global para as técnicas baseadas na Tensão de Pico em janela fixa, divergindo completamente do cenário de cruzamento por zero. Como a perturbação é inserida no exato instante em que a forma de onda atinge o seu valor máximo positivo (90°), o algoritmo de pico registra o valor de 1,0 pu logo antes ou no primeiro instante do início da falta. Consequentemente, a função matemática de máximo absoluto mantém o pico saudável armazenado na memória durante todo o restante daquela janela de cálculo, tornando o afundamento transitoriamente imperceptível.

O detector só é capaz de sinalizar a anomalia na janela subsequente, quando todas as amostras contidas no bloco já pertencem à região degradada do sinal. A Figura 19 demonstra esse atraso crítico nos métodos de pico fixo, evidenciando a perda de confiabilidade temporal desse algoritmo para faltas incidentes em regiões de máximo valor instantâneo.

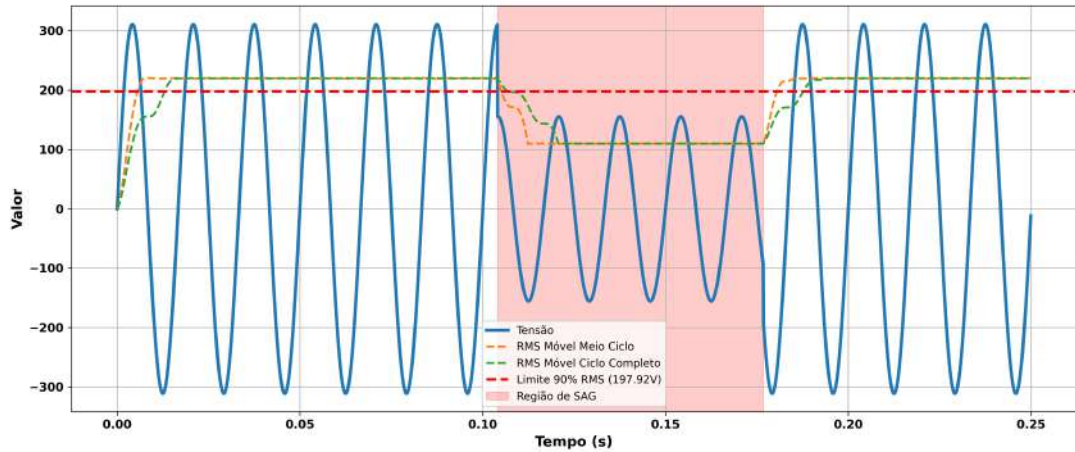
Figura 19 – Comportamento da detecção utilizando métodos de Tensão de Pico (Cenário 90°).

Fonte: Elaborado pelo autor.

Paralelamente, os métodos baseados em RMS Móvel mantêm um desempenho al-

tamente responsivo. Como a janela de integração deslizante elimina a dependência de blocos estáticos e recebe amostras com valores significativamente reduzidos na região de máxima energia (onde a área sob a curva deveria ser maior), o valor eficaz estimado decai rapidamente de forma contínua, como pode ser observado na Figura 20.

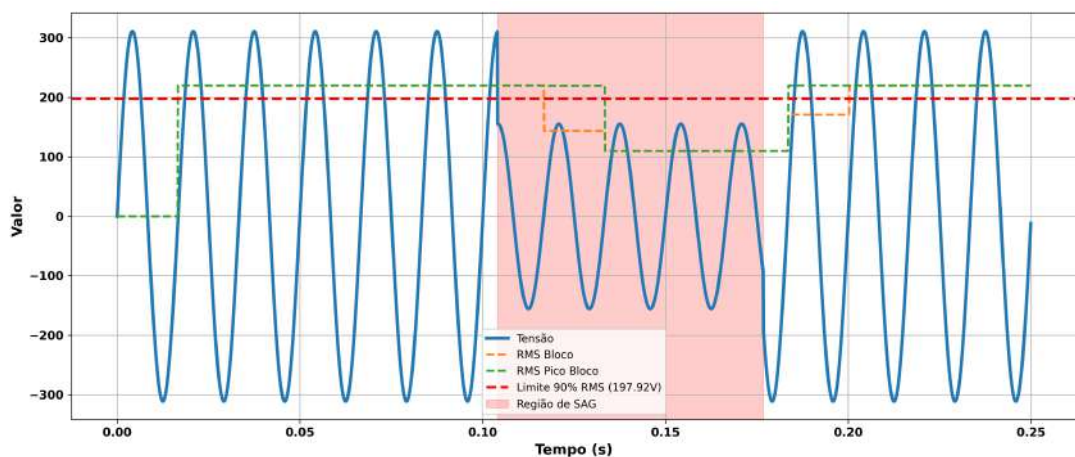
Figura 20 – Comportamento da detecção utilizando janelas de RMS Móvel (Cenário 90°).



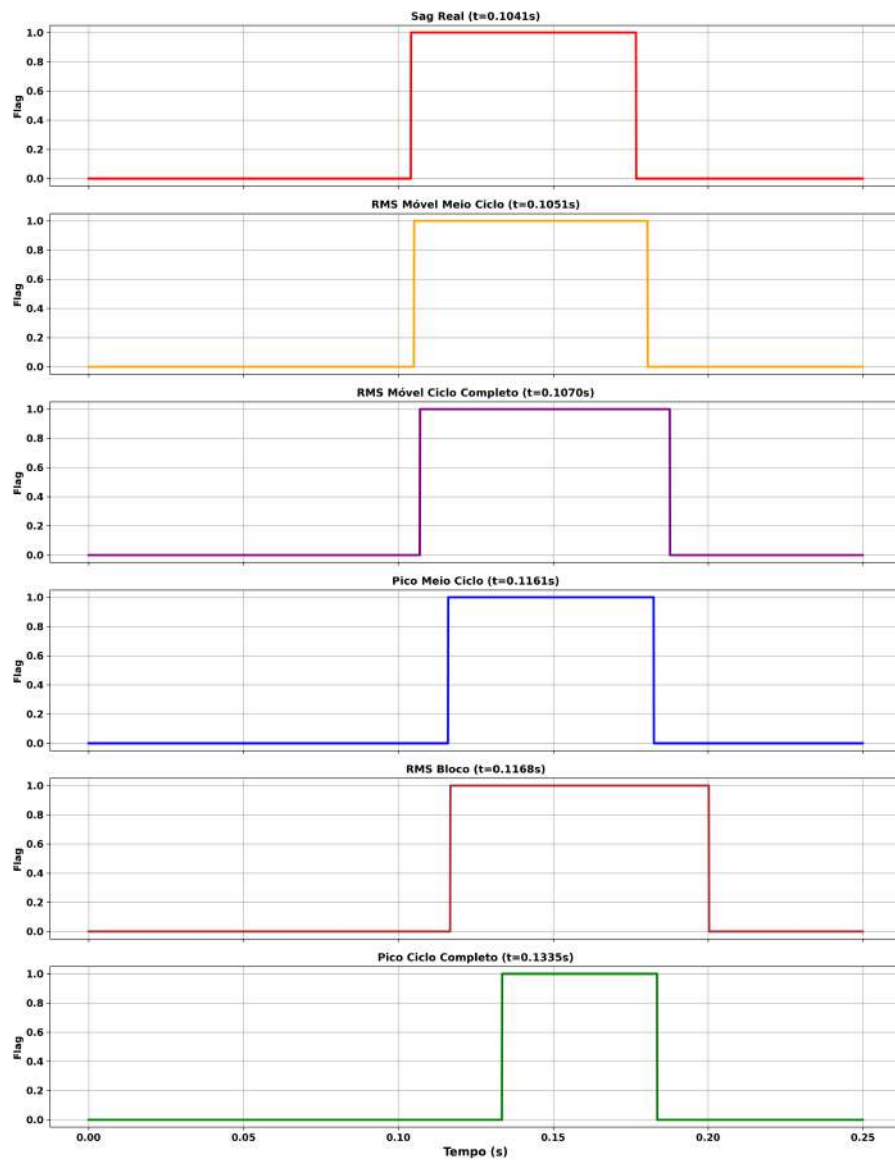
Fonte: Elaborado pelo autor.

A resposta em blocos, tanto para o cálculo RMS quanto para o cálculo de Pico, continua dependente do instante em que a janela de amostragem é encerrada, o que os mantém como os métodos mais lentos, conforme evidenciado na Figura 21. A resposta booleana das *flags* de detecção do sistema é apresentada na Figura 22.

Figura 21 – Comportamento da detecção utilizando cálculo RMS em Bloco vs Pico em Bloco (Cenário 90°).

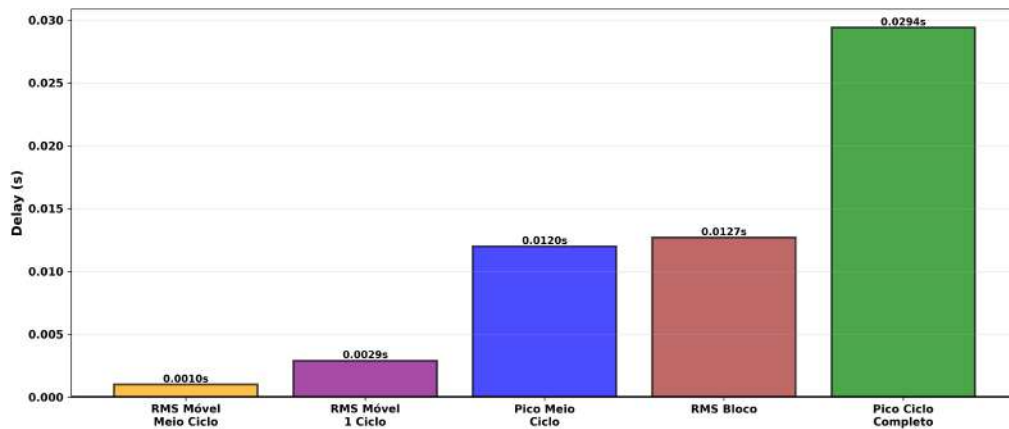


Fonte: Elaborado pelo autor.

Figura 22 – Sinalização (*flags*) de detecção dos algoritmos (Cenário 90°).

Fonte: Elaborado pelo autor.

A avaliação quantitativa do tempo de latência, ilustrada pela Figura 23 e sumariada na Tabela 4, comprova a melhoria no tempo de resposta para grande parte dos métodos devido a este ângulo de falta.

Figura 23 – Atraso de detecção para cada algoritmo no Cenário 3 (90°).

Fonte: Elaborado pelo autor.

Tabela 4 – Resumo do atraso de detecção no Cenário 3 (90°).

Método	Atraso de detecção (ms)
RMS Móvel (Meio Ciclo)	1,0
RMS Móvel (1 Ciclo)	2,9
Pico (Meio Ciclo)	12,0
RMS Bloco	12,7
Pico (1 Ciclo Completo)	29,4

Fonte: Elaborado pelo autor.

Estes resultados demonstram que, embora os métodos de RMS Móvel continuem sendo os mais rápidos (1,0 ms e 2,9 ms), o comportamento dos algoritmos baseados em preenchimento total de janela e de pico sofre forte oscilação, não garantindo robustez na proteção da rede.

4.1.4 Efeito da magnitude do afundamento na latência de detecção

Para além do ângulo de incidência (*Point-on-Wave*), testes adicionais foram conduzidos para avaliar a influência da profundidade do afundamento sobre a velocidade de detecção. O limite de segurança do sistema foi mantido em 0,9 pu. Manteve-se o pior caso angular para métodos móveis (falta em 0°, $t = 0,1000$ s) e simulou-se uma anomalia menos severa, reduzindo o sinal nominal para 80% de sua amplitude (0,8 pu), contrastando com a queda agressiva para 50% avaliada anteriormente.

Os resultados obtidos para a falta de 0,8 pu estão sumarizados na Tabela 5.

Tabela 5 – Atraso de detecção para afundamento leve (0,8 pu) no cruzamento por zero (0°).

Método	Atraso de detecção (ms)
RMS Móvel (Meio Ciclo)	4,2
Pico (Meio Ciclo)	7,8
RMS Móvel (1 Ciclo)	10,1
RMS Bloco	16,8
Pico (1 Ciclo Completo)	16,8

Fonte: Elaborado pelo autor.

Constatou-se que afundamentos menos severos aumentam o tempo de reação dos métodos baseados em integração (RMS). O método RMS Móvel de Meio Ciclo, que detectou o afundamento para 0,5 pu em 3,0 ms neste mesmo ângulo, precisou de 4,2 ms para sinalizar o evento de 0,8 pu. Contudo, mesmo sofrendo este incremento em virtude da menor variação de energia do sinal, o RMS Móvel de Meio Ciclo continuou apresentando o menor tempo de resposta geral entre as metodologias testadas, reafirmando sua eficiência.

4.1.5 Síntese da validação em simulação

O conjunto de simulações com variação do ângulo de incidência da falta (0°, 45° e 90°) e variação da magnitude do afundamento permitiu comprovar a dependência que certos algoritmos possuem em relação às características do evento no instante da perturbação.

Métodos processados em blocos (RMS e Pico em janelas fixas) apresentaram latências extremamente variáveis que podem atingir atrasos de até um ciclo completo da rede elétrica, o que é indesejável em sistemas de monitoramento de cargas sensíveis.

Por outro lado, as técnicas baseadas em janelas deslizantes (RMS Móvel) demonstraram alta imunidade ao ângulo da falta, atualizando a estimativa amostra a amostra. Dentre todos os métodos avaliados, o **RMS Móvel de Meio Ciclo** apresentou a melhor relação entre confiabilidade e velocidade de resposta, registrando atrasos consistentes entre 1,0 ms e 3,0 ms em todos os cenários severos (0,5 pu), além de manter a liderança em tempo de resposta para faltas leves (0,8 pu).

Desta forma, os resultados computacionais fundamentaram a escolha exclusiva do algoritmo de RMS Móvel de Meio Ciclo para ser implementado no sistema embarcado.

4.2 IMPLEMENTAÇÃO EM SISTEMA EMBARCADO

4.2.1 Otimização computacional do RMS Móvel

A implementação de algoritmos de janelas deslizantes em sistemas embarcados exige atenção ao custo computacional, visto que os cálculos são executados a cada novo período de amostragem. Uma implementação direta do método RMS móvel exigiria, a

cada nova amostra adquirida, o deslocamento de todo o vetor de histórico na memória e o recálculo do somatório de N amostras elevadas ao quadrado. Para uma janela de 167 amostras, isso representaria centenas de operações aritméticas, elevando substancialmente a carga de processamento da CPU.

Para otimizar este processo e viabilizar a execução em tempo real, adotou-se a técnica de *buffer* circular aliada ao algoritmo de soma em execução (*running sum*). Nesta abordagem, o somatório dos quadrados das tensões da iteração atual (S_k) é obtido aproveitando o somatório da iteração anterior (S_{k-1}), subtraindo o quadrado da amostra mais antiga que sai da janela (V_{k-N}^2) e adicionando o quadrado da nova amostra recém-adquirida (V_k^2), conforme a equação:

$$S_k = S_{k-1} - V_{k-N}^2 + V_k^2 \quad (4.1)$$

Desta forma, a complexidade matemática do somatório é reduzida de $O(N)$ para $O(1)$, tornando o tempo de execução constante e independente do tamanho da janela de amostragem. A implementação dessa lógica no *firmware* foi realizada através da atualização contínua da variável de soma, seguida pelo cálculo da raiz quadrada. O uso de um ponteiro iterativo que retorna a zero ao atingir o limite N implementou o comportamento de *buffer* circular no vetor de histórico de amostras, eliminando a necessidade de deslocamento físico dos dados na memória RAM do microcontrolador.

4.2.2 Configuração de hardware e processamento em tempo real

Para garantir a fidelidade do processamento em relação aos resultados obtidos na simulação computacional, o microcontrolador STM32F767 foi configurado para operar com uma frequência de *clock* de 216 MHz. A aquisição do sinal de tensão foi realizada pelo conversor analógico-digital (ADC) interno, configurado com resolução de 12 *bits*.

Para assegurar uma taxa de amostragem determinística de 10 kHz, utilizou-se o temporizador de *hardware* TIM2, configurado para gerar gatilhos (*triggers*) de conversão a cada 100 μs . As amostras convertidas foram transferidas para a memória RAM via DMA, configurado para atuar de forma contínua.

A execução do algoritmo matemático de RMS Móvel de Meio Ciclo ocorre integralmente dentro da rotina de interrupção de término de conversão do DMA. A escolha pela otimização da soma em execução, descrita na seção anterior, foi crucial nesta etapa: como a interrupção ocorre a cada 100 μs , todo o cálculo de magnitude, atualização de *buffers* circulares e verificação de limiares precisava ser executado em uma fração desse tempo para evitar a sobrecarga computacional (*bottleneck*). A combinação do *clock* de 216 MHz com a complexidade constante $O(1)$ do algoritmo permitiu processar cada amostra com ampla margem de segurança temporal.

4.2.3 Registro de eventos com RTC

Para o rastreamento cronológico dos eventos de afundamento, o sistema utilizou o periférico interno de *Real-Time Clock* (RTC). Durante a execução contínua da amostragem, caso o valor RMS calculado pelo algoritmo deslizante fique abaixo do limiar de detecção estipulado, uma *flag* booleana de afundamento é acionada na lógica de controle. Neste exato instante, o sistema captura a data e a hora atuais fornecidas pelo RTC, registrando a marca temporal (*timestamp*) de início da anomalia.

Enquanto a *flag* permanece ativa, o microcontrolador mensura a duração contínua do evento. Assim que a tensão se recupera e ultrapassa o limite predefinido, a *flag* é desativada, a duração final da falta em segundos é consolidada e as informações são transferidas para uma estrutura de histórico de dados em memória.

4.2.4 Validação experimental em hardware

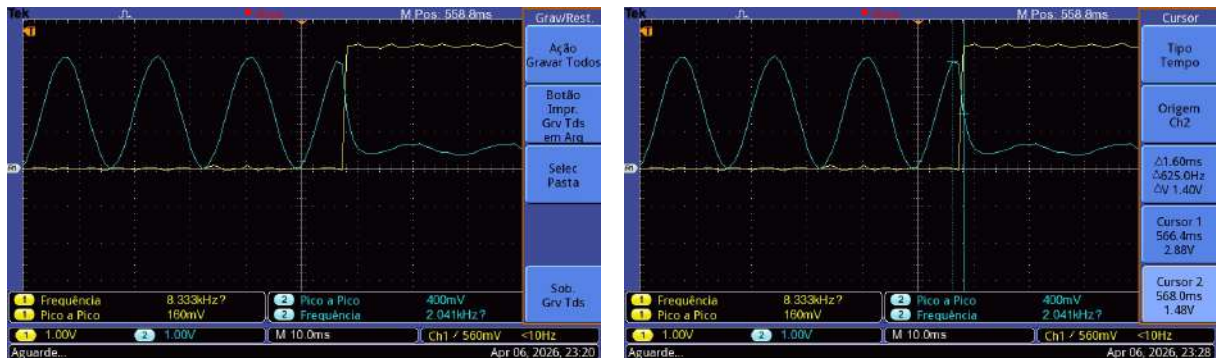
Para atestar a confiabilidade do sistema embarcado e verificar se o tempo de resposta físico coincidia com os dados da simulação teórica, realizou-se uma validação prática utilizando instrumentos de bancada. O arranjo experimental consistiu na aplicação do sinal com afundamento, gerado artificialmente, diretamente no pino de entrada analógica do STM32.

Para a medição rigorosa da latência do sistema, utilizou-se um osciloscópio digital de dois canais. O Canal 2, cuja forma de onda é apresentada em ciano, foi conectado ao sinal senoidal de entrada, permitindo monitorar a tensão real aplicada ao ADC. Já o Canal 1, cuja forma de onda é exibida em amarelo, foi conectado a um pino digital (*GPIO*) configurado no *firmware* para realizar a transição de nível lógico baixo para alto (3,3 V) exatamente no instante em que o algoritmo de RMS Móvel de Meio Ciclo acionasse a *flag* de detecção de afundamento.

A Figura 24 apresenta as capturas de tela do osciloscópio durante a ocorrência de um afundamento iniciado próximo à região de pico da forma de onda.

Figura 24 – Validação prática do tempo de detecção utilizando osciloscópio digital.

- (a) Comportamento da *flag* (CH1) frente ao afundamento (CH2). (b) Medição do atraso de detecção por cursores ($\Delta t = 1,60$ ms).



Fonte: Elaborado pelo autor.

Conforme detalhado na medição por cursores (Figura 24b), o atraso temporal verificado entre o início da queda de tensão e a borda de subida do pino digital de alerta foi de aproximadamente 1,60 ms.

Este resultado prático valida com alta precisão os dados obtidos na simulação computacional, onde faltas iniciadas na região de pico (Cenários 2 e 3) registraram tempo de detecção teórico em torno de 1,0 ms. O acréscimo fracionário de aproximadamente 0,6 ms observado na bancada é inerente ao processamento físico do *hardware*, englobando o tempo de conversão do periférico ADC, o tráfego de dados no barramento via DMA, os ciclos de máquina exigidos pela CPU para calcular a raiz quadrada e a capacitância parasita no acionamento físico da porta GPIO.

Mesmo com as restrições físicas do silício, a detecção ocorreu em menos de 10% do tempo de um ciclo fundamental da rede de 60 Hz, confirmando que a redução de complexidade de $O(N)$ para $O(1)$ foi suficiente para garantir o processamento dentro do intervalo de amostragem de $100 \mu\text{s}$ e a aplicabilidade do microcontrolador na proteção e monitoramento de sistemas elétricos em tempo real.

4.3 VISUALIZAÇÃO DOS EVENTOS DETECTADOS EM INTERFACE WEB

Para prover acesso remoto às informações de qualidade de energia registradas pelo equipamento, desenvolveu-se uma interface de monitoramento baseada em páginas web, hospedada diretamente no microcontrolador.

A infraestrutura de comunicação de rede foi implementada utilizando a pilha de protocolos *Lightweight IP* (LwIP). A integração ao projeto foi realizada por meio do STM32CubeMX, que gerou os arquivos de configuração e inicialização da pilha, incluindo a habilitação do módulo Ethernet do STM32F767 e a alocação dos buffers de recepção e transmissão. O sistema foi configurado para operar como cliente DHCP (*Dynamic Host*

Configuration Protocol), permitindo a obtenção automática de um endereço IP na rede local.

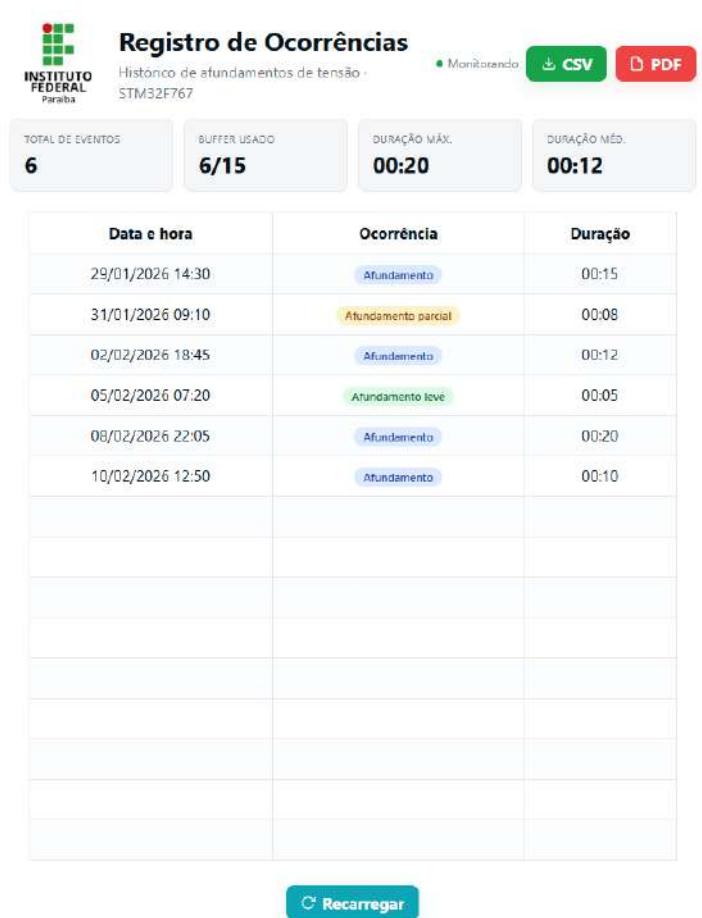
O desenvolvimento da interface visual foi realizado em HTML e CSS, estruturado em um arquivo principal com extensão `.shtml`. A utilização dessa extensão é obrigatória para que o servidor HTTP do LwIP reconheça o arquivo como candidato ao processamento de *Server Side Includes* (SSI). Os demais recursos estáticos, como folhas de estilo, foram organizados em subdiretórios dentro da pasta do projeto de *front-end*.

Para armazenar as páginas na memória *Flash* do microcontrolador, utilizou-se a ferramenta `makefsdata`, disponibilizada nativamente pelo ecossistema STM32 e acessível nas pastas de instalação do STM32CubeIDE, no caminho correspondente às bibliotecas do *middleware* LwIP. Trata-se de um script Perl executado via linha de comando, que recebe como entrada o diretório raiz do *front-end* e gera um arquivo `fsdata.c` contendo os vetores de *bytes* correspondentes a cada página e recurso do projeto. Cada arquivo é representado com seu caminho de URL codificado, permitindo que o servidor HTTP embarcado o localize por nome. O arquivo `fsdata.c` gerado foi então incluído no projeto do STM32CubeIDE e compilado junto ao *firmware*.

A atualização dinâmica dos dados na página foi implementada por meio do mecanismo SSI. No arquivo HTML, foram inseridos marcadores identificadores para cada campo da tabela de ocorrências, correspondendo à data, hora e duração de cada evento. No *firmware*, implementou-se uma função de *callback* que é acionada pelo servidor HTTP sempre que a página é requisitada. Essa função identifica cada marcador pelo seu índice e substitui seu conteúdo pela string correspondente armazenada no histórico de eventos em memória RAM. Os marcadores foram declarados em um vetor de strings no *firmware*, na mesma ordem em que aparecem no arquivo HTML, de forma que o índice recebido pelo *callback* corresponda diretamente ao campo a ser preenchido.

O código HTML da página foi configurado para recarregar automaticamente a cada 1 segundo, de modo que as ocorrências registradas pelo sistema sejam exibidas de forma atualizada ao usuário. A Figura 25 apresenta a interface desenvolvida com o painel de ocorrências preenchido com as marcações de tempo capturadas pelo RTC embarcado.

Figura 25 – Interface web desenvolvida para monitoramento do histórico de afundamentos de tensão.



Fonte: Elaborado pelo autor.

5 DISCUSSÃO

Neste capítulo são discutidos os aspectos técnicos e analíticos extraídos dos resultados da avaliação dos métodos de detecção de afundamento de tensão. A análise aprofunda o comportamento dos modelos matemáticos frente à variação do ângulo de incidência da falta (*Point-on-Wave*) e sua magnitude, bem como o desempenho observado na transposição teórica para a viabilidade prática em um sistema embarcado de tempo real.

5.1 INFLUÊNCIA DA NATUREZA MATEMÁTICA, ÂNGULO DE INCIDÊNCIA E MAGNITUDE

Os resultados obtidos nas simulações comprovam que a latência de detecção diverge significativamente dependendo da formulação matemática do algoritmo, do instante exato em que a perturbação atinge a rede e da severidade da queda de tensão.

Uma das descobertas mais relevantes foi a comprovação do caráter não determinístico dos algoritmos processados em blocos fixos (RMS e Pico em Ciclo Completo). A estimativa da magnitude nesses métodos é atualizada apenas ao final do preenchimento da janela de N amostras, tornando o tempo de resposta dependente da coincidência entre o início da falta e o término do bloco, comportamento indesejável em sistemas de proteção elétrica.

Os métodos baseados em Tensão de Pico demonstraram alta vulnerabilidade e dependência em relação ao ângulo de incidência da falta. Ao contrário do que se projeta intuitivamente, uma falha iniciada em 90° (pico da onda) resulta no pior cenário de latência para os algoritmos de bloco fixo, pois o valor máximo nominal é capturado no início do intervalo de amostragem, mascarando o distúrbio durante todo o ciclo corrente e elevando o tempo de resposta para 29,4 ms. Essa característica inviabiliza o uso isolado da tensão de pico em bloco para sistemas de proteção que exigem comportamento determinístico.

Em contraste, as metodologias baseadas em janelas móveis demonstraram baixa sensibilidade à variação angular. O RMS Móvel de Meio Ciclo confirmou-se como a técnica de maior consistência, com tempos de detecção entre 1,0 ms e 3,0 ms em todos os cenários avaliados para faltas a 50%.

Esses resultados dialogam com os obtidos por (Naidoo; Pillay, 2007), que propuseram um filtro adaptativo não linear reportando latências entre 1 ms e 4 ms, com 9 ms para RMS de um ciclo e 20 ms para tensão de pico, valores coerentes com os 16,7 ms e até 29,4 ms obtidos neste trabalho, diferenças atribuíveis ao contexto distinto de amostragem. O RMS Móvel de Meio Ciclo atingiu desempenho comparável ao filtro adaptativo daqueles autores, porém com complexidade $\mathcal{O}(1)$ via *running sum*, tornando-o uma alternativa atrativa para implementação em microcontroladores sem sistema operacional.

O trabalho de (Sillapawicharn, 2015) propõe um detector ortogonal baseado em

diferenciador, reportando tempo médio de 0,33 ms e pior caso de 0,88 ms para afundamentos de 0,5 pu, desempenho superior ao obtido neste trabalho. Contudo, o diferenciador amplifica componentes de alta frequência, aumentando a sensibilidade a ruídos e harmônicos, limitação reconhecida pelo próprio autor. O método aqui implementado, por operar exclusivamente via *running sum* de complexidade $\mathcal{O}(1)$, oferece maior robustez em arquiteturas *bare-metal*.

Complementarmente, (Guan *et al.*, 2024) aborda afundamentos assimétricos com harmônicos de ordem ímpar, combinando o método de atraso de pequeno ângulo com filtro de média móvel de meio ciclo, com atraso total de aproximadamente 11 ms. Esse cenário evidencia uma limitação do presente trabalho: o RMS Móvel de Meio Ciclo, por operar com janela reduzida de $M = N/2$ amostras, é potencialmente mais sensível a distorções harmônicas, o que não se manifesta aqui por o sistema operar com sinal senoidal controlado. A extensão para ambientes com harmônicos constitui uma direção natural para trabalhos futuros.

Por fim, constatou-se que afundamentos de menor magnitude (0,8 pu) demandam tempos de detecção levemente maiores, pois a convergência da média móvel abaixo do limiar de 0,9 pu exige mais iterações quando a variação de energia por amostra é menor. Ainda assim, o RMS Móvel de Meio Ciclo manteve o melhor desempenho dentre todos os métodos avaliados.

5.2 CONSIDERAÇÕES SOBRE O DESEMPENHO EM TEMPO REAL

A transposição do ambiente ideal de simulação para a arquitetura de um microcontrolador impõe restrições físicas severas, tornando aspectos como custo algorítmico, carga de processamento (*CPU Load*) e latência de memória, fatores limitantes para algoritmos de Processamento Digital de Sinais.

A escolha do método RMS Móvel de Meio Ciclo apresentou duas vantagens principais. Além do desempenho matemático verificado na simulação, sua adaptação lógica permitiu contornar as limitações computacionais do microcontrolador STM32F767. A implementação do conceito de *buffer* circular aliado ao algoritmo de soma em execução (*running sum*) reduziu a complexidade das operações matemáticas iterativas de $O(N)$ para $O(1)$.

Sem esta otimização, o processamento de uma janela deslizante exigiria um laço de repetição com dezenas de multiplicações e somas além do deslocamento de vetores em memória, o que poderia sobrecarregar a Unidade Lógica e Aritmética (ALU) do processador, visto que o temporizador exigia a resolução de toda a matemática em um intervalo crítico inferior a $100 \mu s$.

Adicionalmente, a delegação da captura dos dados do ADC para o periférico de acesso direto à memória (DMA) demonstrou ser uma decisão arquitetural fundamental.

O DMA absorveu o custo temporal da amostragem, permitindo que a CPU de 216 MHz do STM32 se dedicasse inteiramente à resolução da lógica otimizada $O(1)$ na rotina de interrupção, garantindo um determinismo temporal que validou a operação em tempo real do sistema.

5.3 VIABILIDADE AUTÔNOMA E CONECTIVIDADE

A última consideração diz respeito à capacidade do sistema em atuar como um nó autônomo de monitoramento. A implementação de uma interface web embarcada não serviu apenas para validação visual, mas demonstrou que microcontroladores modernos possuem capacidade estrutural para operar de forma equivalente a analisadores de qualidade de energia convencionais, porém com custos e dimensões físicas substancialmente reduzidos.

A adoção do periférico RTC interno conferiu rigor cronológico aos eventos, garantindo que o *timestamp* da falha pudesse ser correlacionado com outros relatórios industriais. Ademais, a utilização da pilha LwIP com DHCP demonstrou a viabilidade de dispositivos de monitoramento *plug and play*, que dispensam configurações complexas de rede.

Por fim, a conversão das páginas web para código binário via `makefsdata` e a injeção dinâmica de variáveis com o protocolo SSI provaram ser técnicas altamente eficientes para contornar a ausência de um sistema operacional de alto nível (*bare-metal*). Essa abordagem descentraliza o monitoramento de distúrbios elétricos, permitindo que os distúrbios detectados pelo algoritmo de média móvel sejam acessados por qualquer operador conectado à rede.

6 CONCLUSÃO

Este trabalho apresentou o desenvolvimento, a avaliação e a implementação prática de um sistema autônomo para detecção de afundamentos de tensão (*voltage sag*). O objetivo central do estudo foi expandir a análise comparativa de algoritmos matemáticos, demonstrando a viabilidade de implementação de técnicas de processamento digital de sinais (PDS) de alta velocidade em arquiteturas baseadas em microcontroladores.

A primeira etapa consistiu na modelagem e simulação computacional de cinco algoritmos de detecção, avaliados sob a influência do ângulo de incidência da falta (*Point-on-Wave*) em 0° , 45° e 90° , bem como sob diferentes níveis de severidade (0,5 pu e 0,8 pu). Os resultados evidenciaram limitações importantes dos métodos clássicos. Observou-se que algoritmos baseados em blocos fixos apresentam comportamento não determinístico, podendo introduzir atrasos de até um ciclo completo da rede elétrica (16,7 ms). De forma complementar, verificou-se que os métodos baseados na detecção de tensão de pico em bloco apresentam elevada dependência do ângulo de incidência da falta, resultando em severa perda de sensibilidade e atrasos críticos (atingindo 29,4 ms) quando o evento se inicia no pico da forma de onda (90°), uma vez que o valor máximo nominal é retido na memória e mascara o distúrbio pelo restante do ciclo.

Em contrapartida, os métodos baseados em janelas deslizantes apresentaram elevada robustez frente à variação angular. Dentre as técnicas avaliadas, o método de **RMS Móvel de Meio Ciclo** destacou-se por sua estabilidade e rapidez, apresentando tempos de detecção consistentes na faixa de 1,0 ms a 3,0 ms para afundamentos severos (0,5 pu). Além disso, manteve desempenho superior mesmo em afundamentos de menor magnitude (0,8 pu), nos quais a dinâmica de convergência da média móvel naturalmente exige maior número de amostras para atingir o limiar de detecção.

A etapa de implementação embarcada confirmou a viabilidade prática dos resultados obtidos em simulação. O sistema foi desenvolvido utilizando o microcontrolador STM32F767, no qual o algoritmo de RMS Móvel foi otimizado por meio da utilização de um *buffer* circular e da técnica de soma em execução, reduzindo a complexidade computacional de $O(N)$ para $O(1)$. Em conjunto com o uso do controlador de acesso direto à memória (DMA) para aquisição de dados, foi possível garantir processamento determinístico dentro do intervalo de amostragem de $100 \mu s$, sem introduzir gargalos computacionais na unidade de processamento.

Além da detecção dos eventos, o sistema demonstrou capacidade de operação como um dispositivo autônomo de monitoramento. A integração do *Real-Time Clock* (RTC) permitiu o registro temporal preciso dos eventos, enquanto a utilização da pilha de protocolos LwIP associada à técnica de *Server Side Includes* (SSI) viabilizou a disponibilização dos dados por meio de uma interface web embarcada. Essa abordagem possibilita o acesso remoto às informações de forma simples, utilizando apenas um navegador conectado à

rede.

Como limitação deste trabalho, destaca-se que os ensaios foram realizados em ambiente controlado, com sinais predominantemente senoidais, não contemplando condições mais representativas de ambientes industriais reais, como presença significativa de distorções harmônicas ou ruídos elétricos. Além disso, a avaliação experimental foi conduzida em escala de bancada, o que indica a necessidade de validações adicionais em ambientes reais de sistemas elétricos.

Como trabalhos futuros, sugere-se primeiramente a validação do sistema em condições reais de operação, substituindo o gerador de sinais por um circuito de condicionamento conectado diretamente à rede elétrica, utilizando sensores como o módulo ZMPT101B ou transformadores de potencial, expondo o algoritmo a condições práticas como harmônicos, ruídos de chaveamento e afundamentos de origem externa.

Adicionalmente, recomenda-se a ampliação das funcionalidades de detecção para outros distúrbios classificados pelo PRODIST Módulo 8 e pela norma IEEE 1159, como sobretensões (swells), interrupções momentâneas e análise de distorção harmônica total (THD). Por fim, a integração com protocolos de comunicação industrial, como MQTT e Modbus TCP, permitiria a inserção do dispositivo em arquiteturas de monitoramento distribuído no contexto da Indústria 4.0.

REFERÊNCIAS

- AGÊNCIA NACIONAL DE ENERGIA ELÉTRICA (ANEEL). **Procedimentos de Distribuição de Energia Elétrica no Sistema Elétrico Nacional: PRODIST – Módulo 8**. 10. ed. S.l.: ANEEL, 2017.
- DECKMANN, Sigmar Maurer; POMILIO, José Antenor. **Qualidade da energia elétrica: conceituação e processamento digital**. Editora Blucher, 2024.
- GUAN, Xin *et al.* Asymmetric Voltage Sag Detection Method Based on Delay Small Angle Method and Moving Average Filter. *In: 2024 IEEE 4th New Energy and Energy Storage System Control Summit Forum (NEESSC)*. 2024. p. 34–41. DOI: 10.1109/NEESSC62857.2024.10733428.
- IEEE Recommended Practice for Monitoring Electric Power Quality. **IEEE Std 1159-2019 (Revision of IEEE Std 1159-2009)**, p. 1–98, 2019. DOI: 10.1109/IEEESTD.2019.8796486.
- LATHI, B. P. **Sinais e Sistemas Lineares**. 2. ed. Porto Alegre: Bookman, 2005.
- MARAFÃO, Fernando Pinhabel. Análise e controle da energia elétrica através de técnicas de processamento digital de sinais. **Universidade Estadual de Campinas-UNICAMP, Campinas, Tese de Doutorado**, 2004.
- MARTINS, Júlio S; COUTO, Carlos; AFONSO, João Luiz. Qualidade de energia eléctrica. **3o Congr. Luso-Moçambicano Eng.–CLME’2003 Eng. e Inovação para o Desenvolv**, p. 219–231, 2003.
- MAZURKIEWICZ, Greg. **Power Outages Cost \$119 Billion per Year**. Acesso em: 2026-04-11. 2001. Disponível em: <https://www.achrnews.com/articles/98926-power-outages-cost-119-billion-per-year>.
- NAIDOO, Raj; PILLAY, Pragasen. A New Method of Voltage Sag and Swell Detection. **IEEE Transactions on Power Delivery**, v. 22, n. 2, p. 1056–1063, 2007. DOI: 10.1109/TPWRD.2007.893185.
- OPPENHEIM, Alan V.; WILLSKY, Alan S.; YOUNG, Ian T. **Sinais e Sistemas**. 2. ed. São Paulo: Prentice Hall, 2002.
- PINKERTON. **Energy Resilience Solutions: Combatting Power Outages and Economic Loss**. 2025. Disponível em: <https://pinkerton.com/our-insights/blog/energy-resilience-solutions-combatting-power-outages-and-economic-loss>. Acesso em: 12 abr. 2026.
- RADHAKRISHNA, C.; ESHWARDAS, M.; CHEBIYAM, G. Impact of voltage sags in practical power system networks. *In: 2001 IEEE/PES Transmission and Distribution*

Conference and Exposition. Developing New Perspectives (Cat. No.01CH37294). 2001. v. 1, 567–572 vol.1. DOI: 10.1109/TDC.2001.971296.

RIBEIRO, Moisés Vidal. **Técnicas de processamento de sinais aplicadas a transmissão de dados via rede elétrica e ao monitoramento da qualidade de energia**. 2005. Tese (Doutorado) – University of Campinas, Brazil.

SANTOS, Fábio. **Tecnologias IoT na Segurança Industrial**. Dez. 2020. Tese (Doutorado).

SBQEE, V. Impacto econômico do afundamento de tensão na indústria: uma metodologia aplicada a Grandes redes elétricas, 2003.

SILLAPAWICHARN, Yutthachai. A fast voltage sag detector based on peak detection. *In*: 2015 12th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON). 2015. p. 1–4. DOI: 10.1109/ECTICon.2015.7206952.

STMICROELECTRONICS. **STM32F76xxx and STM32F77xxx Advanced ARM-based 32-bit MCUs**. Jul. 2024. Acesso em: 2026-04-13. Disponível em: https://www.st.com/resource/en/reference_manual/rm0410-stm32f76xxx-and-stm32f77xxx-advanced-armbased-32bit-mcus-stmicroelectronics.pdf.

STMICROELECTRONICS. **NUCLEO-F767ZI Development Board**. 2016. Acesso em: 13 Apr. 2026. Disponível em: <https://www.st.com/en/evaluation-tools/nucleo-f767zi.html>.

STMICROELECTRONICS. **STM32CubeIDE User Manual**. 2021. Acesso em: 05 mar. 2025. Disponível em: https://www.st.com/resource/en/user_manual/um2609-stm32cubeide-user-guide-stmicroelectronics.pdf.

STMICROELECTRONICS. **UM1713: Developing applications on STM32Cube with LwIP TCP/IP stack**. Rev 4. 2016. Acesso em: 19 Apr. 2026. Disponível em: https://www.st.com/resource/en/user_manual/um1713-developing-applications-on-stm32cube-with-lwip-tcpip-stack-stmicroelectronics.pdf.

TUNABOYLU, N.S.; COLLINS, E.R.; CHANEY, P.R. Voltage disturbance evaluation using the missing voltage technique. *In*: 8TH International Conference on Harmonics and Quality of Power. Proceedings (Cat. No.98EX227). 1998. v. 1, 577–582 vol.1. DOI: 10.1109/ICHQP.1998.759974.

ZHANG, Xiao-Ping; YAN, Zuanhong. Energy Quality: A Definition. **IEEE Open Access Journal of Power and Energy**, v. 7, p. 430–440, 2020. DOI: 10.1109/OAJPE.2020.3029767.

ANEXO A – ANÁLISE DE AFUNDAMENTOS

Este anexo apresenta o código fonte desenvolvido em linguagem C utilizado para simular e calcular os algoritmos de detecção de afundamentos de tensão através dos métodos RMS móvel de um ciclo, RMS móvel de meio ciclo, RMS em bloco, e os picos de meio ciclo e ciclo completo.

Listagem A.1 – Código em C para processamento e análise de afundamentos.

```

1 #include <stdio.h>
2 #include <math.h>
3 #include <stdbool.h>
4
5 #define FS 10000.0f
6 #define FREQ 60.0f
7 #define N 167
8 #define N_MEIO (N/2)
9 #define DURACAO 0.25f
10
11 #define PI 3.14159265358979323846f
12 float period[N] = {0};
13 float period_meio[N_MEIO] = {0};
14
15 /* ===== RMS MOVEL 1 CICLO ===== */
16 float calcular_rms_movel(float amostra, float *soma, int indice)
17 {
18     float novo = amostra * amostra;
19     float antigo = period[indice] * period[indice];
20
21     *soma -= antigo;
22     *soma += novo;
23
24     return sqrtf(*soma / N);
25 }
26
27 /* ===== RMS MOVEL MEIO CICLO ===== */
28 float calcular_rms_movel_meio(float amostra, float *soma, int indice)
29 {
30     float novo = amostra * amostra;
31     float antigo = period_meio[indice] * period_meio[indice];
32
33     *soma -= antigo;
34     *soma += novo;
35
36     return sqrtf(*soma / N_MEIO);
37 }
38
39 /* ===== RMS BLOCO ===== */
40 float calcular_rms_bloco(float amostra, float *soma,
41                         int *contador, float rms_ant)
42 {
43     *soma += amostra * amostra;
44     (*contador)++;
45
46     if(*contador >= N)

```

```
47     {
48         float rms = sqrtf(*soma / N);
49         *soma = 0;
50         *contador = 0;
51         return rms;
52     }
53
54     return rms_ant;
55 }
56
57 /* ===== MAIN ===== */
58 int main()
59 {
60     FILE *fp = fopen("dados_rms.csv", "w");
61     if(!fp)
62     {
63         printf("Erro ao criar CSV\n");
64         return 1;
65     }
66
67     fprintf(fp, "tempo,tensao,rms_movel,rms_movel_meio,rms_bloco,"
68             "pico_meio,pico_ciclo,"
69             "sag_real,flag_movel,flag_meio_movel,"
70             "flag_bloco,flag_pico_meio,flag_pico_ciclo\n");
71
72     float Ts = 1.0f / FS;
73     float T_ciclo = 1.0f / FREQ;
74     int total = (int)(DURACAO * FS);
75
76     float soma_movel=0, soma_meio=0, soma_bloco=0;
77
78     float rms_movel=0, rms_meio_movel=0, rms_bloco=0;
79     float rms_pico_meio=0, rms_pico_ciclo=0;
80
81     float pico_meio=0, pico_ciclo=0;
82
83     int i=0, i_meio=0;
84     int cont_bloco=0;
85     int cont_meio=0, cont_ciclo=0;
86
87     float amplitude=311.0f;
88     float rms_nominal=amplitude/sqrtf(2.0f);
89     float limite=0.9f*rms_nominal;
90
91     float inicio_sag=0.10f;
92     float fim_sag=0.16f;
93
94     int flag_movel=0, flag_meio_movel=0;
95     int flag_bloco=0, flag_pico_meio=0, flag_pico_ciclo=0;
96
97     float t_det_movel=-1, t_det_meio_movel=-1;
98     float t_det_bloco=-1, t_det_pico_meio=-1, t_det_pico_ciclo=-1;
99
100    for(int k=0;k<total;k++)
101    {
102        float tempo=k*Ts;
```

```
104     int sag_real=0;
105     float amp=amplitude;
106
107     if(tempo>=inicio_sag && tempo<=fim_sag)
108     {
109         amp*=0.5f;
110         sag_real=1;
111     }
112
113     float amostra=amp*sinf(2*PI*REQ*tempo);
114
115     /* ===== RMS MOVEL 1 CICLO ===== */
116     rms_movel=calcular_rms_movel(amostra,&soma_movel,i);
117
118     if(k>N && rms_movel<limite)
119     {
120         if(!flag_movel) t_det_movel=tempo;
121         flag_movel=1;
122     }
123     else flag_movel=0;
124
125     period[i]=amostra;
126
127     /* ===== RMS MOVEL MEIO CICLO ===== */
128     rms_meio_movel=calcular_rms_movel_meio(amostra,&soma_meio,i_meio);
129
130     if(k>N_MEIO && rms_meio_movel<limite)
131     {
132         if(!flag_meio_movel) t_det_meio_movel=tempo;
133         flag_meio_movel=1;
134     }
135     else flag_meio_movel=0;
136
137     period_meio[i_meio]=amostra;
138
139     /* ===== RMS BLOCO ===== */
140     rms_bloco=calcular_rms_bloco(amostra,&soma_bloco,&cont_bloco,rms_bloco);
141
142     if(k>N && rms_bloco<limite)
143     {
144         if(!flag_bloco) t_det_bloco=tempo;
145         flag_bloco=1;
146     }
147     else flag_bloco=0;
148
149     /* ===== PICO MEIO ===== */
150     float abs_val=fabsf(amostra);
151
152     if(abs_val>pico_meio) pico_meio=abs_val;
153     cont_meio++;
154
155     if(cont_meio>=N_MEIO)
156     {
157         rms_pico_meio=pico_meio/sqrtf(2.0f);
158
159         if(k>N && rms_pico_meio<limite)
160         {
```

```
161         if(!flag_pico_meio) t_det_pico_meio=tempo;
162         flag_pico_meio=1;
163     }
164     else flag_pico_meio=0;
165
166     pico_meio=0;
167     cont_meio=0;
168 }
169
170 /* ===== PICO CICLO ===== */
171 if(abs_val>pico_ciclo) pico_ciclo=abs_val;
172 cont_ciclo++;
173
174 if(cont_ciclo>=N)
175 {
176     rms_pico_ciclo=pico_ciclo/sqrtf(2.0f);
177
178     if(k>N && rms_pico_ciclo<limite)
179     {
180         if(!flag_pico_ciclo) t_det_pico_ciclo=tempo;
181         flag_pico_ciclo=1;
182     }
183     else flag_pico_ciclo=0;
184
185     pico_ciclo=0;
186     cont_ciclo=0;
187 }
188
189 fprintf(fp, "%.6f,%.6f,%.6f,%.6f,%.6f,%.6f,%.6f,%d,%d,%d,%d,%d,%d\n",
190         tempo, amostra,
191         rms_movel, rms_meio_movel, rms_bloco,
192         rms_pico_meio, rms_pico_ciclo,
193         sag_real,
194         flag_movel, flag_meio_movel,
195         flag_bloco, flag_pico_meio, flag_pico_ciclo);
196
197 i++; if(i>=N) i=0;
198 i_meio++; if(i_meio>=N_MEIO) i_meio=0;
199 }
200
201 fclose(fp);
202
203 printf("\n===== RESULTADOS =====\n");
204 printf("Sag iniciou: %.6f s\n\n", inicio_sag);
205
206 printf("RMS MOVEL 1 CICLO: %.6f s\n", (t_det_movel));
207 printf("RMS MOVEL MEIO CICLO: %.6f s\n", (t_det_meio_movel));
208 printf("RMS BLOCO: %.6f s\n", (t_det_bloco));
209 printf("PICO MEIO CICLO: %.6f s\n", (t_det_pico_meio));
210 printf("PICO CICLO: %.6f s\n", (t_det_pico_ciclo));
211
212 printf("\nCSV gerado com sucesso.\n");
213 return 0;
214 }
```

ANEXO B – INTERFACE WEB (STM32)

Este anexo documenta o código estrutural em HTML e folha de estilos CSS embarcado na memória do microcontrolador STM32F767, desenvolvido para exibir o painel e o histórico de registros de afundamentos de tensão monitorados em tempo real.

Listagem B.1 – Estrutura HTML e CSS da interface web residencial no microcontrolador.

```

1 <!doctype html>
2 <html lang="pt-br">
3   <head>
4     <meta charset="utf-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1" />
6     <title>Registro de Ocorrências</title>
7     <style>
8       :root {
9         color-scheme: light;
10        --bg: #f4f6f8;
11        --card: #ffffff;
12        --text: #06121a;
13        --muted: #6b7280;
14        --accent: #0ea5b7;
15        --border: #e6e9ee;
16        --amber-bg: #fef3c7;
17        --amber-text: #92400e;
18        --blue-bg: #dbeafe;
19        --blue-text: #1e40af;
20        --green-bg: #dcfce7;
21        --green-text: #166534;
22      }
23
24      * { box-sizing: border-box; }
25
26      body {
27        margin: 0;
28        font-family: system-ui, -apple-system, Segoe UI, Roboto, Ubuntu,
29        Cantarell, "Helvetica Neue", Arial, sans-serif;
30        background: var(--bg);
31        color: var(--text);
32        min-height: 100vh;
33        display: grid;
34        place-items: center;
35        padding: 1rem 0;
36      }
37
38      .container {
39        background: var(--card);
40        padding: 2.25rem 1.25rem;
41        border-radius: 10px;
42        text-align: center;
43        max-width: 860px;
44        width: 96%;
45        box-shadow: 0 6px 20px rgba(16,24,40,0.06);
46        border: 1px solid var(--border);
47      }


```

```
48     h1 { margin: 0 0 0.25rem; font-size: 1.6rem; }
49
50     .subtitle { margin: 0; font-size: 0.9rem; color: var(--muted); }
51
52     p { margin: 0; line-height: 1.5; }
53
54     .brand {
55         display: flex;
56         align-items: center;
57         gap: 12px;
58         justify-content: space-between;
59         margin-bottom: 14px;
60     }
61     .logo { width: 96px; height: auto; display: block; }
62     .brand-text { text-align: left; }
63
64     .status-pill {
65         display: inline-flex;
66         align-items: center;
67         gap: 6px;
68         font-size: 12px;
69         color: var(--muted);
70         margin-top: 6px;
71     }
72     .status-dot {
73         width: 7px;
74         height: 7px;
75         border-radius: 50%;
76         background: #16a34a;
77         flex-shrink: 0;
78     }
79
80     .stats {
81         display: grid;
82         grid-template-columns: repeat(4, 1fr);
83         gap: 10px;
84         margin: 1rem 0 1.25rem;
85         text-align: left;
86     }
87
88     .stat-card {
89         background: var(--bg);
90         border: 1px solid var(--border);
91         border-radius: 8px;
92         padding: 0.75rem 0.875rem;
93     }
94
95     .stat-card .label {
96         font-size: 10px;
97         color: var(--muted);
98         text-transform: uppercase;
99         letter-spacing: 0.05em;
100        margin-bottom: 4px;
101    }
102
103    .stat-card .value {
104        font-size: 1.35rem;
```

```
105     font-weight: 700;
106     color: var(--text);
107     font-variant-numeric: tabular-nums;
108     min-height: 1.6rem;
109   }
110
111   .warn-banner {
112     display: block;
113     min-height: 0;
114     padding: 10px 14px;
115     border-radius: 8px;
116     background: var(--amber-bg);
117     border: 1px solid #fcd34d;
118     margin-bottom: 1rem;
119     font-size: 12px;
120     color: var(--amber-text);
121     text-align: left;
122     font-weight: 600;
123   }
124
125   table { width: 100%; border-collapse: collapse; margin-bottom: 1.5rem; }
126
127   th, td {
128     border: 1px solid var(--border);
129     padding: 0.5rem 0.6rem;
130     text-align: center;
131     color: var(--text);
132     font-size: 0.95rem;
133   }
134
135   table.records { max-width: 760px; margin: 0 auto 1.5rem; }
136
137   table.records thead th {
138     background: transparent;
139     color: var(--text);
140     font-weight: 700;
141     padding: 0.6rem 0.8rem;
142   }
143
144   tbody tr td { height: 40px; }
145   tbody tr:nth-child(odd) td { background: #fbfdff; }
146
147   .badge {
148     display: inline-block;
149     font-size: 11px;
150     font-weight: 600;
151     padding: 2px 9px;
152     border-radius: 999px;
153   }
154   .badge-sag      { background: var(--blue-bg); color: var(--blue-text); }
155   .badge-partial { background: var(--amber-bg); color: var(--amber-text); }
156   .badge-light   { background: var(--green-bg); color: var(--green-text); }
157
158   .actions {
159     display: flex;
160     align-items: center;
161     justify-content: center;
```

```
162     gap: 8px;
163     flex-wrap: wrap;
164     margin-top: 8px;
165   }
166
167   .btn {
168     display: inline-flex;
169     align-items: center;
170     padding: 0.45rem 0.9rem;
171     border-radius: 8px;
172     background: var(--accent);
173     color: #fff;
174     text-decoration: none;
175     font-weight: 700;
176     box-shadow: 0 6px 18px rgba(14,165,183,0.12);
177     border: none;
178     cursor: pointer;
179     font-size: inherit;
180     font-family: inherit;
181     gap: 6px;
182   }
183   .btn:active { transform: translateY(1px); }
184
185   .last-update { font-size: 11px; color: var(--muted); margin-top: 10px; }
186
187   @media (max-width: 720px) {
188     .stats { grid-template-columns: repeat(2, 1fr); }
189     .brand { gap: 8px; flex-direction: column; align-items: center; }
190     .logo { width: 72px; }
191     .container { padding: 1rem; }
192     h1 { font-size: 1.2rem; }
193   }
194 </style>
195 </head>
196 <body>
197   <main class="container">
198     <header class="brand">
199       
200       <div class="brand-text">
201         <h1>Registro de Ocorrências</h1>
202         <p class="subtitle">Histórico de afundamentos de tensão · STM32F767</p>
203       </div>
204       <div class="status-pill">
205         <span class="status-dot"></span>
206         <span>Monitorando</span>
207       </div>
208     </header>
209
210     <div class="stats">
211       <div class="stat-card">
212         <div class="label">Total de eventos</div>
213         <div class="value"><!--#total--></div>
214       </div>
215       <div class="stat-card">
216         <div class="label">Buffer usado</div>
217         <div class="value"><!--#bufused--></div>
```

```
218     </div>
219     <div class="stat-card">
220         <div class="label">Duração máx.</div>
221         <div class="value"><!--#maxdur--></div>
222     </div>
223     <div class="stat-card">
224         <div class="label">Duração méd.</div>
225         <div class="value"><!--#avgdur--></div>
226     </div>
227 </div>
228
229 <div class="warn-banner"><!--#warn--></div>
230
231 <table class="records">
232     <thead>
233         <tr>
234             <th>Data e hora</th>
235             <th>Ocorrência</th>
236             <th>Duração</th>
237         </tr>
238     </thead>
239     <tbody>
240         <tr><td><!--#dt0--></td><td><!--#oc0--></td><td><!--#dur0--></td></tr>
241         <tr><td><!--#dt1--></td><td><!--#oc1--></td><td><!--#dur1--></td></tr>
242         <tr><td><!--#dt2--></td><td><!--#oc2--></td><td><!--#dur2--></td></tr>
243         <tr><td><!--#dt3--></td><td><!--#oc3--></td><td><!--#dur3--></td></tr>
244         <tr><td><!--#dt4--></td><td><!--#oc4--></td><td><!--#dur4--></td></tr>
245         <tr><td><!--#dt5--></td><td><!--#oc5--></td><td><!--#dur5--></td></tr>
246         <tr><td><!--#dt6--></td><td><!--#oc6--></td><td><!--#dur6--></td></tr>
247         <tr><td><!--#dt7--></td><td><!--#oc7--></td><td><!--#dur7--></td></tr>
248         <tr><td><!--#dt8--></td><td><!--#oc8--></td><td><!--#dur8--></td></tr>
249         <tr><td><!--#dt9--></td><td><!--#oc9--></td><td><!--#dur9--></td></tr>
250     </tbody>
251 </table>
252
253 <div class="actions">
254     <a class="btn" href="/index.shtml">Recarregar</a>
255 </div>
256
257 <p class="last-update"><!--#lastupd--></p>
258 </main>
259 </body>
260 </html>
```

	INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
	Campus João Pessoa - Código INEP: 25096850
	Av. Primeiro de Maio, 720, Jaguaribe, CEP 58015-435, João Pessoa (PB)
	CNPJ: 10.783.898/0002-56 - Telefone: (83) 3612.1200

Documento Digitalizado Ostensivo (Público)

Trabalho de Conclusão de Curso - Corrigido (Já com Ficha Catalográfica e Folha de Aprovação).

Assunto:	Trabalho de Conclusão de Curso - Corrigido (Já com Ficha Catalográfica e Folha de Aprovação).
Assinado por:	Lucas Medeiros
Tipo do Documento:	Anexo
Situação:	Finalizado
Nível de Acesso:	Ostensivo (Público)
Tipo do Conferência:	Cópia Simples

Documento assinado eletronicamente por:

- **Lucas Medeiros Carneiro Costa, DISCENTE (20211610040) DE BACHARELADO EM ENGENHARIA ELÉTRICA - JOÃO PESSOA**, em 06/07/2026 14:23:25.

Este documento foi armazenado no SUAP em 06/07/2026. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 1904313

Código de Autenticação: be42828dd3

