

IMPLEMENTAÇÃO EM MÓDULO DE DESENVOLVIMENTO FPGA DE TÉCNICA DE DETECÇÃO DE NOVIDADES BASEADA DA TRANSFORMADA DE STOCKWELL APLICADA A DISTÚRBIOS DE QUALIDADE DE ENERGIA ELÉTRICA EM SINAIS DE POTÊNCIA

IMPLEMENTATION IN FPGA DEVELOPMENT KIT OF NOVELTY DETECTION TECHNIQUE BASED ON STOCKWELL TRANSFORM APPLIED TO POWER QUALITY DISTURBANCES IN POWER SIGNALS

Anonmixs Amaro Afonso¹
Naiara Da Silva Maia Dos Santos²
Victor Mendes Ribeiro³
Eder Barboza Kapisch⁴

DOI: 10.34019/2179-3700.2024.v24.46211

ENVIADO EM: 8/10/2024

APROVADO EM: 11/11/2024

RESUMO

Diante do cenário de consolidação das Redes Elétricas Inteligentes (REI), existe um grande potencial para o aparecimento de novos distúrbios elétricos. Além disso, considerando a enorme quantidade de dados provenientes dos medidores inteligentes, faz-se necessária a utilização de métodos de detecção de novidades nos sinais de forma de onda de tensão e corrente, a fim de preservar as informações relevantes e promover

¹Universidade Federal de Juiz de Fora, campus Juiz de Fora, Minas Gerais, Brasil. Discente do curso de Engenharia Elétrica – Sistemas Eletrônicos. Programa VIC/UFJF, <https://orcid.org/0009-0003-7643-2248>. E-mail: anonmixs.afonso@estudante.ufjf.br.

²Universidade Federal de Juiz de Fora, campus Juiz de Fora, Minas Gerais, Brasil. Discente do curso de Engenharia Elétrica – Sistemas Eletrônicos. Programa VIC/UFJF, <https://orcid.org/0009-0007-5684-4711>. E-mail: naiara.maia@estudante.ufjf.br.

³Universidade Federal de Juiz de Fora, campus Juiz de Fora, Minas Gerais, Brasil. Discente do Programa de Pós-graduação em Engenharia Elétrica (PPEE) da UFJF, <https://orcid.org/0009-0005-2865-019X>. E-mail: victormrmat@gmail.com.

⁴Professor Adjunto da Universidade Federal de Juiz de Fora. Orientador do projeto de IC. Doutor e Engenharia Elétrica pelo Programa de Pós-graduação em Engenharia Elétrica (PPEE) da UFJF. Membro do grupo de pesquisa em Processamento de Sinais Aplicado a Sistemas de Potência (PSCOPE), <https://orcid.org/0000-0001-7609-7943>. Endereço profissional: Universidade Federal de Juiz de Fora (UFJF), campus Juiz de Fora, Faculdade de Engenharia, Departamento Circuitos Elétricos, São Pedro, Rua José Lourenço Kelmer, CEP: 36.036-900 - Juiz de Fora, MG – Brasil. E-mail: eder.kapisch@ufjf.br.

um armazenamento eficiente da informação. A Transformada de Stockwell (*Stockwell Transform* - ST) é uma distribuição tempo-frequência que tem mostrado grande capacidade de detectar novidades relacionadas a quebras de estacionariedade no sinal. Sendo assim, o presente trabalho descreve a utilização da ST para fins de detecção de novidades em sinais de Qualidade de Energia Elétrica (QEE). A implementação dessa transformada em *Field-Programmable Gate Array* (FPGA) é proposta através da utilização de um processador *soft-core* para otimizar os recursos de *hardware* da FPGA. Além disso, é proposta uma estratégia de seleção de vozes, a fim de diminuir a complexidade do algoritmo, bem como diminuir o tempo de processamento, mantendo-se a capacidade de detecção.

Palavras-chave: Redes Elétricas Inteligentes (REI). Transformada de Stockwell (ST). Qualidade de Energia Elétrica (QEE). Implementação em FPGA. Detecção de Novidades (DN).

ABSTRACT

In the context of the consolidation of Smart Grids (SGs), there is a substantial potential for the emergence of new disturbances. Additionally, given the vast amount of data from smart meters, it is necessary to use novelty detection methods on voltage and current waveform signals to preserve relevant information and enable efficient data storage. The Stockwell Transform (ST) is a time-frequency distribution that has shown a strong ability to detect novelties related to signal stationarity changes. Therefore, this work describes the use of the ST for novelty detection in Power Quality (PQ) signals. The implementation of this transform on a Field-Programmable Gate Array (FPGA) platform is proposed through the use of a soft-core processor to optimize the FPGA hardware resources. Furthermore, a voice selection strategy is proposed to reduce algorithm complexity and processing time while maintaining detection capability.

Keywords: Smart Grids (SG). Stockwell Transform (ST). Power Quality (PQ); FPGA Implementation. Novelty Detection (ND).

1 INTRODUÇÃO

O crescimento dos Sistemas Elétricos de Potência (SEP) e a introdução de novas cargas não lineares tornaram a rede elétrica mais complexa e suscetível a distúrbios, observáveis nas formas de onda de corrente e tensão. Com os avanços tecnológicos na informática e no processamento dos dispositivos de medição e controle, surgiram as Redes Elétricas Inteligentes (REI), também conhecidas como *Smart Grids* (SG) (KATARAY, 2023).

As REI visam atender à demanda por gerenciamento eficiente dos recursos elétricos, permitindo que o sistema opere de forma autônoma, detectando e mitigando problemas de Qualidade de Energia Elétrica (QEE) em tempo real. Para identificar novos tipos de distúrbios, as REI precisam preservar todas as

informações relevantes nas ondas de tensão e corrente, uma tarefa desafiadora devido ao volume crescente de dados gerados pelos dispositivos de medição em tempo real (DE OLIVEIRA e BOLLEN, 2023).

Com o aumento no número de dispositivos inteligentes monitorando grandezas elétricas, a quantidade de dados a ser armazenada pode ser expressiva. Assim, é essencial utilizar técnicas de processamento de sinais que possibilitem um armazenamento eficiente desses dados (KAPISCH *et al.*, 2021). Nesse contexto, técnicas de detecção de novidades desempenham um papel importante.

Entre as técnicas promissoras para a detecção de novidades, destaca-se a Transformada de Stockwell (ST), foco deste presente artigo. Ela possui propriedades teórico-matemáticas significativas e uma alta capacidade de detectar quebras de estacionariedade (STOCKWELL *et al.*, 1996). Sua utilização tem se tornado cada vez mais comum na literatura. A seguir, alguns trabalhos correlatos com o tema.

Em SHANG-BIN *et al.* (2018), foi desenvolvida uma versão rápida da ST para analisar as características de energia transitória e o ângulo de fase da corrente de sequência zero. Foram utilizados casos simulados e experimentais para verificar a eficácia do método proposto em termos de precisão e análise em tempo real. Como resultado, houve uma considerável redução da complexidade temporal do algoritmo original da ST, mantendo suas características de tempo-frequência.

Em KUMAR *et al.* (2023), é apresentado um algoritmo online para monitoramento de distúrbios usando a ST e baseado em árvore de decisão. Os sinais analisados foram simulados no MATLAB[®] com diferentes combinações de causas e ruído com Relação Sinal-Ruído (*Signal-to-Noise Ratio* – SNR) de 40 dB, sendo posteriormente testados com sinais reais da rede elétrica.

O trabalho descrito em RIBEIRO *et al.*, (2024) mostra resultados de implementação em módulo de kit de desenvolvimento FPGA do detector de novidades exposto no presente artigo, servindo como uma continuação das pesquisas introduzidas aqui.

A ST tem se destacado como uma técnica eficaz para a detecção de novidades em sinais, apresentando propriedades teórico-matemáticas relevantes e uma elevada capacidade para identificar quebras de estacionariedade, principalmente em aplicações que envolvam sinais de QEE.

O uso kits de desenvolvimento baseados em FPGA pode proporcionar flexibilidade, validação de implementação de técnicas e algoritmos em tempo real e rapidez de prototipação para testes em campo. A tecnologia presente da FPGA tem ganhado destaque nas tarefas de desenvolvimento de circuitos de Processamento Digital de Sinais (PDS), tendo em vista sua capacidade de processamento paralelo, granularidade e altas taxas de *clock* (BALASUBBAREDDY, 2023).

Sendo assim, no presente trabalho optou-se no pela utilização desta tecnologia, com foco na implementação de um detector de novidades baseado na ST, a fim de melhorar a detecção em tempo real, especialmente em sinais variantes no tempo. Além disso, é possível utilizar processadores customizáveis dentro da FPGA (denominados de processadores *soft-core*), buscando a otimização dos recursos de *hardware* disponíveis.

Dessa maneira, o objetivo principal é descrever o desenvolvimento um detector de novidades para aplicação de QEE implementado em FPGA com o auxílio de um processador embarcado, a fim de otimizar os recursos de *hardware*.

O restante do trabalho está organizado da forma como segue. A Seção 2 descreve brevemente a teoria e o desenvolvimento matemático da ST. A Seção 3 traz a metodologia da implementação do detector de novidades proposto em plataforma FPGA. A Seção 4 mostra alguns resultados e discussão. Por fim, a Seção 5 traz algumas conclusões sobre o trabalho.

2 FORMULAÇÃO TEÓRICA

A ST é uma distribuição tempo-frequência que pode ser utilizada para a análise de sinais variantes no tempo. Foi criada por R. G. Stockwell em 1996, a fim de criar uma representação espectral complexa para sinais não estacionários (STOCKWELL *et al.*, 1996). Sua definição pode ser dada por):

$$S(\tau, f) = \int_{-\infty}^{\infty} x(t) \frac{|f|}{\sqrt{2\pi}} e^{-\frac{(t-\tau)^2 f^2}{2}} e^{-i2\pi ft} dt, \quad (1)$$

onde $S(\tau, f)$ é a matriz complexa da transformada de Stockwell com suas dimensões de tempo τ e frequência f e $x(t)$ é o sinal contínuo no tempo a ser

analisado. Dentro do integrando, pode-se observar o termo $\frac{|f|}{\sqrt{2\pi}}$, que representa o valor máximo de amplitude dependente da frequência que o envelope gaussiano possui. Já o termo $e^{-\frac{(t-\tau)^2 f^2}{2}}$ representa a modulação dessa amplitude em forma de distribuição gaussiana, onde pode-se ver o termo τ como um deslocamento temporal do envelope. Finalmente, o termo $e^{-i2\pi ft}$, sendo i a unidade imaginária, é o responsável pela frequência a ser analisada e é o mesmo que dá ao conjunto um caráter complexo, permitindo que a ST contenha informações do módulo e da fase do sinal analisado. Tal termo é o mesmo visto nas bases da transformada de Fourier (TF) (KAPISCH, 2019).

2.1 A superfície do módulo da Transformada de Stockwell

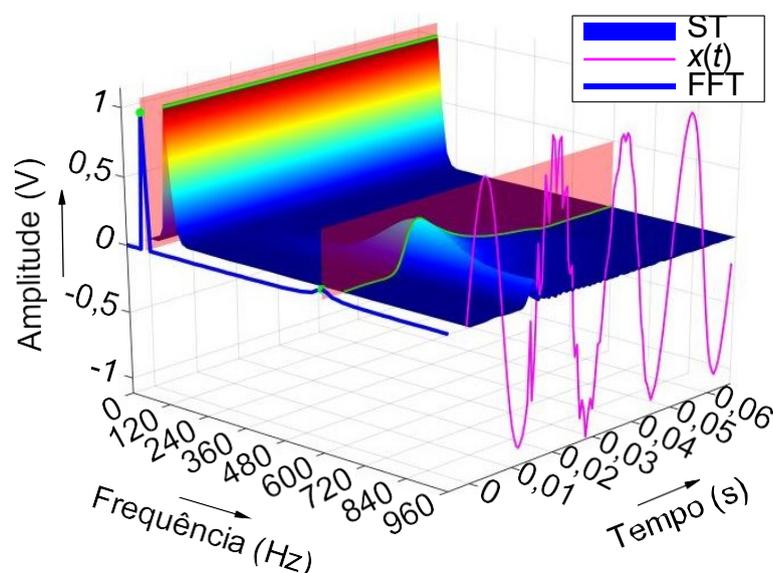
O resultado da ST é uma matriz complexa contendo informações espectrais em relação ao tempo e frequência de um sinal. Uma das informações importantes a serem extraídas dessa matriz é o seu módulo, que serve como estimativa da amplitude das componentes espectrais variantes no tempo presentes no sinal. O módulo da matriz ST fornece uma superfície tridimensional. Ao realizarmos um corte nessa superfície sendo fixado um instante de tempo t_0 , pode-se obter o espectro local para esse instante. Por outro lado, pode-se fazer um corte ao longo do tempo, fixando-se uma determinada frequência f_0 , obtendo-se o que denominamos de *voice* da frequência f_0 . Esse *voice* revela como a componente de frequência f_0 se comporta ao longo do tempo, possibilitando a avaliação de quebras de estacionariedade.

A Figura 1 mostra um exemplo da superfície do módulo da ST. Além disso, é mostrado o sinal $x(t)$ ao longo do tempo na linha de cor magenta, bem como a transformada de Fourier, calculada pela Transformada Rápida Fourier (*Fast Fourier Transform* - FFT) ao longo do domínio da frequência em linha azul. Nesta figura, foram também selecionados dois *voices* relacionados aos valores máximos da FFT, um localizado na frequência fundamental do sinal (60 Hz), outro localizado na frequência do distúrbio transitório (600 Hz).

É possível notar que, se selecionarmos o *voice* localizado em 600 Hz, podemos utilizá-lo para caracterizar a quebra de estacionariedade presente no sinal analisado, indicando que este trecho do sinal possui informação relevante e, portanto, deve ser armazenado.

a. Método de detecção por variância de voices selecionados

O método de detecção de novidades apresentado aqui utiliza a variância dos *voices* extraídos da superfície do módulo da ST. Entretanto, calcular todas as *voices* da superfície do módulo da Transformada de Stockwell para um sinal $x(t)$ com quatro ciclos da frequência fundamental e um distúrbio do tipo transitório.



Fonte: Elaborado pelos próprios autores, 2024.

variâncias de todos os *voices* é uma tarefa computacional que exige uma grande quantidade de processamento. Por isso, em KAPISCH (2019), foi proposta uma estratégia de seleção de *voices*, a qual permite verificar que os bins de maior energia da FFT representam os componentes principais que carregam traços de maior variação dos *voices* da ST.

O presente trabalho implementa essa estratégia em FPGA e mostra que é possível obter um ganho significativo em processamento e tempo de execução, graças à estratégia de seleção de *voices*.

3 METODOLOGIA PROPOSTA

A metodologia adotada para o desenvolvimento do detector de novidades proposto se baseia na utilização da plataforma FPGA, a fim de criar um sistema embarcado dedicado de alta eficiência, devido à sua flexibilidade, alta granularidade, capacidade de processamento e reprogramação dinâmica.

Para a implementação de *hardware* em FPGA de um detector de novidades

baseado na ST através da seleção de *voices* da transformada, foi desenvolvida uma adaptação na transformada para sua programação dentro de um processador *soft-core* embarcado na FPGA. O processador *soft-core* escolhido é denominado *Scalable Architecture Processor for Hardware Optimization* (SAPHO) (RIBEIRO, 2024). Este *soft-core* foi desenvolvido no Núcleo de Instrumentação e Processamento de Sinais da Universidade Federal de Juiz de Fora (NIPS/UFJF), capaz de realizar operações por meio de circuitos aritméticos em ponto fixo e em ponto flutuante. Para validar a implementação, foram realizados testes em laboratório utilizando sinais sintéticos.

Para a geração dos sinais sintéticos de QEE e simulações do algoritmo e dos aprimoramentos propostos na ST, foi utilizado o ambiente MATLAB[®]. Para a descrição do *hardware* e síntese dos blocos na FPGA foi utilizado o sintetizador Quartus Prime[®], da Fabricante Intel[®]. A Linguagem de Descrição de *Hardware* *Hardware Description Language* - HDL) utilizada foi verilog.

3.1 Processador embarcado – SAPHO

O processador *soft-core* SAPHO torna as operações necessárias mais próximas aos algoritmos utilizados na criação da teoria proposta. Para que a FPGA possa receber o processador embarcado, foi criado um bloco, cuja função serve como uma espécie de “berço”, a fim de realizar a interface das saídas e entradas do processador na PFGA.

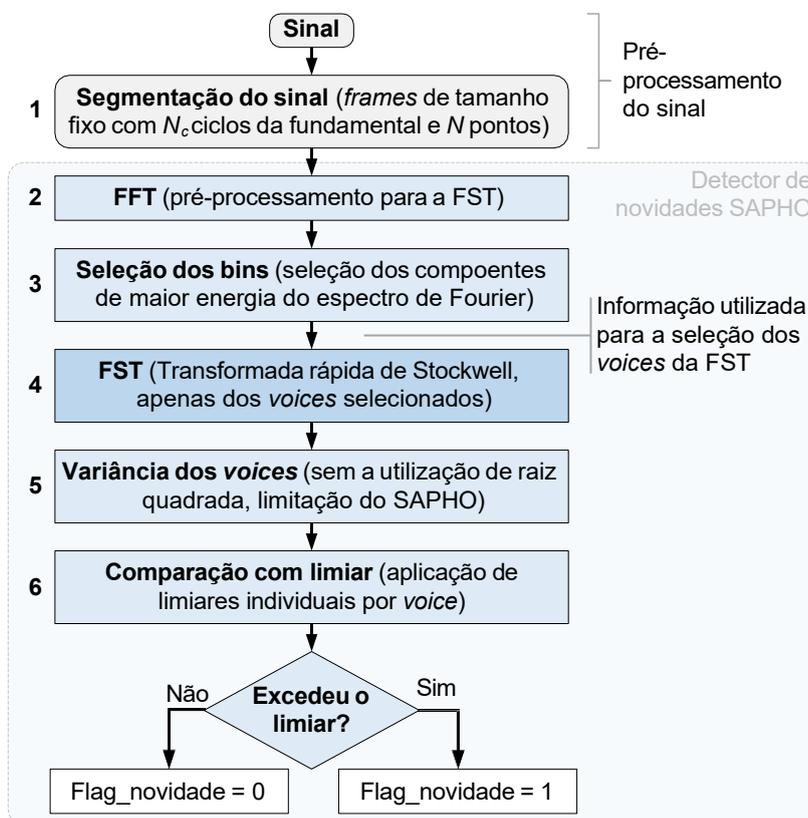
O Ambiente de Desenvolvimento Integrado (*Integrated Development Environment* - IDE) do SAPHO permite que vários parâmetros de projeto sejam customizados, tais como, número de bits da mantissa, expoente, tamanho da pilha de dados, número de saídas, entradas, dentre outros.

Após a implementação do processador e sua interface de dados na FPGA, o processador deve executar os comandos presentes em sua memória de instruções (programa), criados através do IDE do SAPHO que são convertidos instruções em Assembly por um compilador C e em seguida convertidas em código de máquina por um compilador Assembler, ambos integrados ao IDE. A linguagem utilizada para programar o processador SAPHO é formada por um subconjunto da linguagem de programação C, batizada como C+-.

3.2 Implementação do detector de novidades

A etapas percorridas pelo detector de novidades proposto seguem os passos mostrados na Figura 2.

Primeiramente, o sinal é dividido em *frames* de tamanho fixo contendo N_c Figura 2 – Diagrama do bloco de interface do processador embarcado SAPHO.



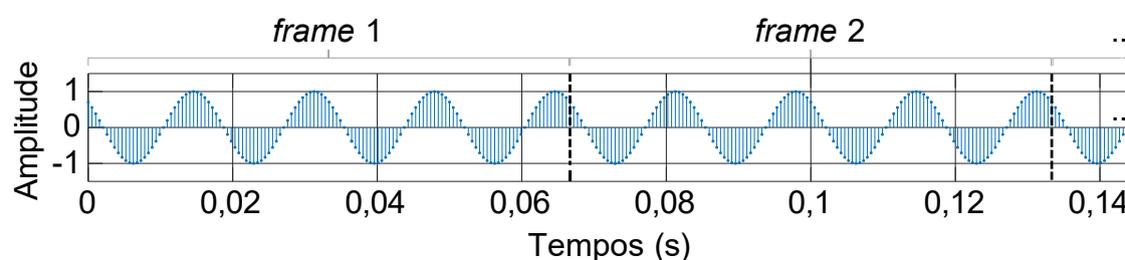
Fonte: Elaborado pelos próprios autores, 2024.

ciclos da componente fundamental. Em seguida, é calculada a FFT, uma vez que a mesma é um pré-processamento para a ST. A partir da FFT, as posições dos máximos são extraídas para o cálculo da Transformada Rápida de Stockwell (*Fast Stockwell Transform*) (FST), que é uma adaptação da ST feita para ser executada no processador embarcado. Depois disso, as variâncias dos *voices* selecionados são calculadas, as quais são avaliadas pela comparação de um limiar pré-estabelecido. Se a variância ultrapassar o limiar, significa que aquele *voice* analisado contém uma novidade e, portanto, o *frame* é considerado um *frame* de novidade e precisa ser armazenado, pois possui informação relevante. Caso todas as variâncias não ultrapassem o limiar, pode-se dizer que o *frame* não possui novidade. Os passos mostrados na Figura 2 são descritos a seguir.

3.2.1 Segmentação

Neste trabalho, adotou-se o tamanho do *frame* de $N_c = 4$ ciclos da frequência fundamental $f_1 = 60$ Hz. Assim, a frequência de amostragem é igual a $F_s = f_1 \times N / N_c = f_1 \times N / 4$. A frequência de amostragem adotada foi de 1920 Hz, ou 32

Figura 3 – Separação do sinal em *frames*.



Fonte: Elaborado pelos próprios autores, 2024.

pontos por ciclo da fundamental, resultando em $N = 128$ pontos por *frame* analisado. A Figura 3 mostra um exemplo de sinal segmentado conforme descrito acima.

3.2.2 Transformada rápida de Fourier (FFT)

Para construir o código da função FFT e IFFT, foi desenvolvido um código em C+- baseado em (PRESS *et al.*, 2003), onde foram necessárias apenas algumas modificações para atender às peculiaridades da linguagem C+- do SAPHO. Nesse código, os parâmetros de entrada e saída da transformada são criados de forma que a ordem dos índices do eixo de frequências é idêntica às simulações criadas no MATLAB®.

3.2.3 Seleção de bins

A fim de selecionar os bins de maior energia da FFT, o algoritmo calcula o módulo da FFT e ordena do maior para o menor. O número de *voices* calculados está relacionado ao número de bins de máxima energia selecionados N_{max} . Devido à característica de esparsidade dos sinais de QEE, adotou-se $N_{max} = 4$ para este trabalho. Isto significa que o número de *voices* selecionados para verificar se há ou não novidades é 4.

3.2.4 Transformada de Stockwell Adaptada (FST)

A fim de tornar a implementação possível e mais eficaz, propõe-se neste trabalho uma adaptação da ST, onde, através de manipulações matemáticas da

Equação), é possível utilizar os recursos das transformadas rápidas FFT e IFFT, obtendo-se a expressão mostrada na Equação (2). Todas as propriedades matemáticas da formulação da ST contínua são preservadas.

$$\mathbf{S}\left(mT, \frac{n}{NT}\right) = \frac{1}{2} F^{-1} \left\{ \underbrace{X\left(F_s k - \frac{n}{NT}\right)}_{=\mathbf{H}(k,n) \text{ "toeplitz"}} \square \mathbf{G}_0\left(2\pi F_s k, \frac{n}{NT}\right) \right\} \quad (2)$$

Em), $\mathbf{S}\left(mT, \frac{n}{NT}\right)$ é a matriz bidimensional complexa resultante do cálculo da ST, cuja primeira dimensão mT possui dimensão temporal e a segunda $\frac{n}{NT}$ possui dimensão de frequência, uma vez que $T = 1/F_s$ é o período de amostragem. $F^{-1}\{\cdot\}$ representa a transformada inversa de Fourier, $X(\cdot)$ é a FFT de $x(t)$ e \square é o produto ponto a ponto das matrizes \mathbf{H} , formada por uma matriz "toeplitz" a partir de $X(\cdot)$ e a matriz \mathbf{G}_0 é definida por:

$$\mathbf{G}_0\left(2\pi F_s k, \frac{n}{NT}\right) = e^{-j\frac{1}{2}\left(\frac{2\pi F_s k}{n}\right)^2 \left(\frac{n}{NT}\right)}, \quad (3)$$

onde $m, k = 0, 1, 2, \dots, N - 1, n = 0, 1, 2, \dots, \frac{N}{2} - 1$.

Considerando uma frequência de amostragem F_s fixa, a matriz \mathbf{G}_0 possui dependência exclusiva do comprimento do *frame* N . Desta forma, pode ser feita a implementação prévia de \mathbf{G}_0 no processador sem a necessidade de ser calculada mais de uma vez. Desta forma, para a implementação da FST, basta a criação de uma função que, ao inicializar o processador, aloque os valores da matriz \mathbf{G}_0 na memória de dados do processador em um *array*, a fim de serem multiplicados ponto a ponto por uma matriz \mathbf{H} "toeplitz" do espectro do sinal.

3.2.5 Transformada de Stockwell Adaptada (FST)

Devido às limitações de operações matemáticas suportadas pelo conjunto de instruções que formam a linguagem C++ do processador, em especial raiz quadrada, foi necessária a utilização de uma base de cálculo diferente para a variância, na qual o módulo do *voice* é elevado ao quadrado, conforme (4).

$$Var_i = \sum_{n=0}^{N-1} \frac{\left(|v_i[n]|^2 - \overline{|v_i|^2} \right)^2}{N}, \tag{4}$$

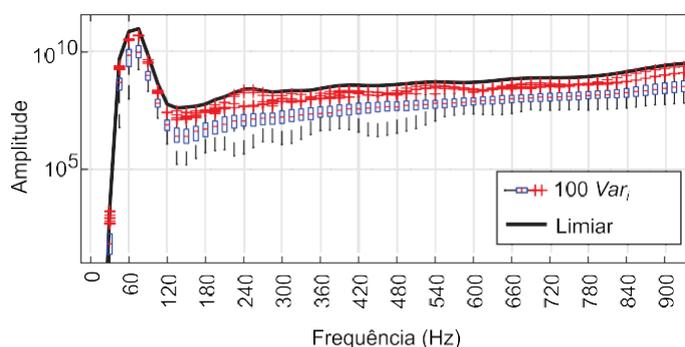
Onde $v_i[n]$ é o vetor de elementos complexos que representa o i -ésimo *voice*, Var_i é a variância do módulo ao quadrado do vetor que representa o *voice* $v_i[n]$ e o traço superior $\overline{\{\cdot\}}$ representa a operação de cálculo do valor médio.

3.2.6 Criação dos limiares

Para criar um detector factível para implementação, é necessário criar um limiar com uma certa flexibilidade, a fim de evitar ocorrências de eventos falsos positivos na detecção. Para isso, foi necessária criação de um banco de sinais que representasse os sinais que não deveriam ser detectados.

Dessa forma, foram criados no MATLAB® 100 *frames* de sinal com amplitude unitária sem distúrbios, corrompidos por um ruído SNR = 30 dB. Foram extraídas as suas variâncias em relação ao tempo de cada *voice* presente em cada sinal. Desta forma, o conjunto formando uma linha de limiares foi criado de modo que fosse superior a todas as variâncias dos *frames* corrompidos. Foi utilizado 3 vezes o desvio padrão das variâncias dos *voices*, conforme mostrado na Figura 4, garantindo que 99,9% das possíveis variâncias dos *frames* corrompidos estejam abaixo do limiar.

Figura 4 – Criação do limiar através desvio padrão.



Fonte: Elaborado pelos próprios autores, 2024.

4 RESULTADOS E DISCUSSÃO

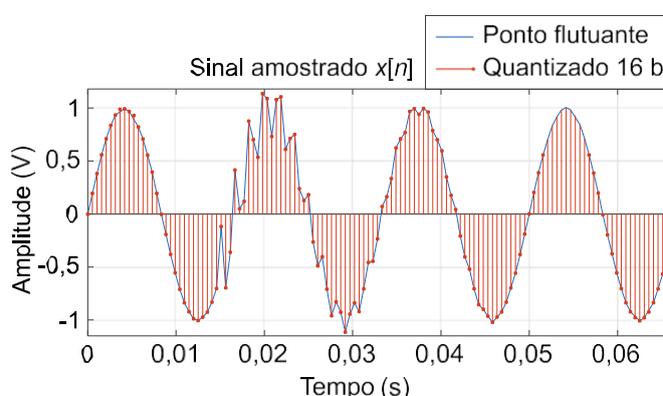
A fim de realizar testes e obter resultados do detector de novidades proposto, duas versões do detector foram criadas para fins comparativos. A primeira utiliza

a FST completa, na qual todos os *voices* são calculados. A segunda versão aplica a estratégia de simplificação baseada na seleção de *voices*. Os resultados obtidos apresentados são relativos à simulação funcional do *hardware* utilizando o ModelSim® comparados com os resultados do MATLAB®.

4.1 FST completa

Para esta primeira simulação, foi utilizado um *frame* com 128 pontos, onde em 0,014 s surge um transitório com frequência de 600 Hz que decai exponencialmente, conforme mostrado na Figura 5.

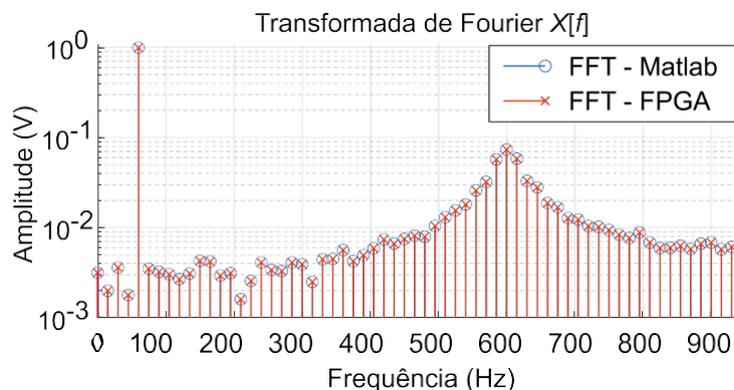
Figura 5 – *Frame* de sinal utilizado na simulação.



Fonte: Elaborado pelos próprios autores, 2024.

Os resultados da FFT aplicada sobre o *frame* em teste efetuada na FPGA podem ser comparados com os resultados obtidos no MATLAB® através da Figura 6. Pode-se ver que os resultados obtidos pelo processador embarcado na FPGA correspondem aos esperados.

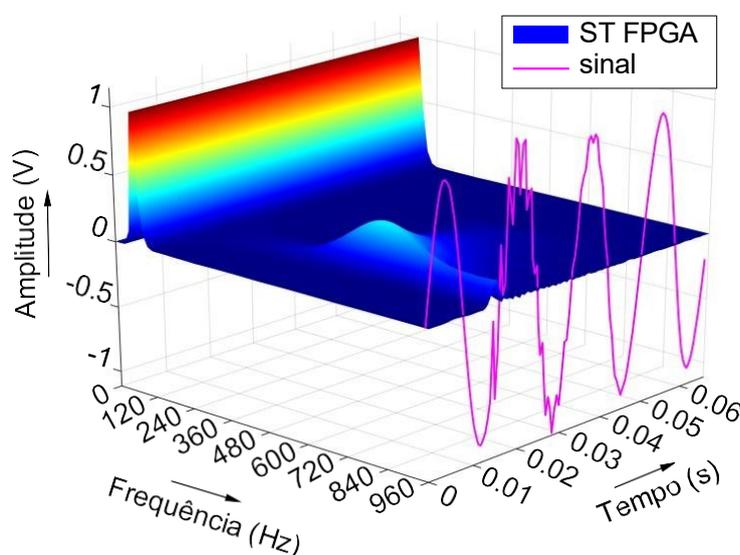
Figura 6 – *Cálculo* da FFT.



Fonte: Elaborado pelos próprios autores, 2024.

Após a comparação dos resultados da FFT, a matriz completa da ST foi calculada pelo processador embarcado. Seu módulo foi computado, fornecendo a superfície tridimensional vista na Figura 7.

O *frame* de sinal aqui analisado foi propositadamente escolhido como o mesmo utilizado no exemplo mostrado na Figura 1, a qual foi obtida pelo Figura 7 – Superfície do módulo da ST calculada pelo SAPHO na FPGA.



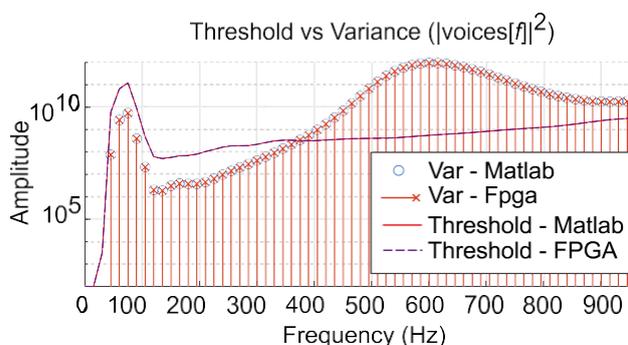
Fonte: Elaborado pelos próprios autores, 2024.

MATLAB®. Podemos ver que as superfícies obtidas foram as mesmas, mostrando a acurácia fornecida pelo processador embarcado.

Pode-se comparar também os resultados da variância obtidos pelo processador em relação aos resultados do MATLAB®.

Podemos notar através da comparação com o limiar, quais são os bins de frequência que possuem novidade através da Figura 8.

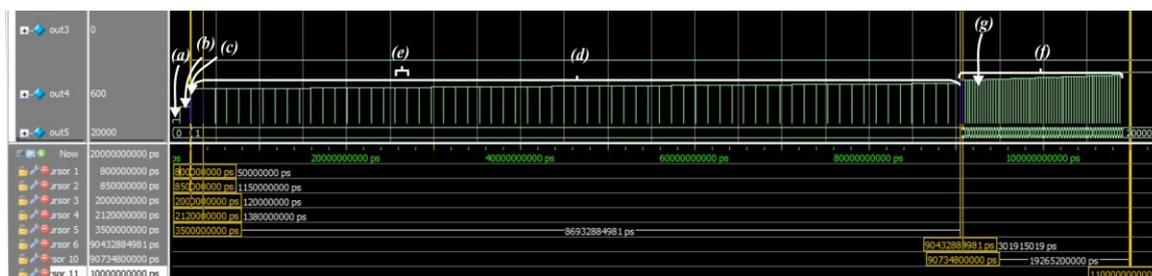
Figura 8 – Comparação do cálculo das variâncias de todos os voices.



Fonte: Elaborado pelos próprios autores, 2024.

A partir do bin de frequência 375 Hz, todos ultrapassam o limiar, mostrando que o *frame* é de novidade. Além disso, podemos ver que a região em torno de 600 Hz possui a maior variância, comprovando que o máximo local do bin de 600 Hz carrega a informação de variação temporal.

Na simulação funcional, foi utilizado um *clock* de 50 MHz e um *frame* Figura 9 – Análise temporal da 1ª simulação.



Fonte: Elaborado pelos próprios autores, 2024.

contendo $N = 128$ pontos. A Figura 9 mostra o diagrama de temporização da simulação funcional do *hardware* implementado para a FST completa.

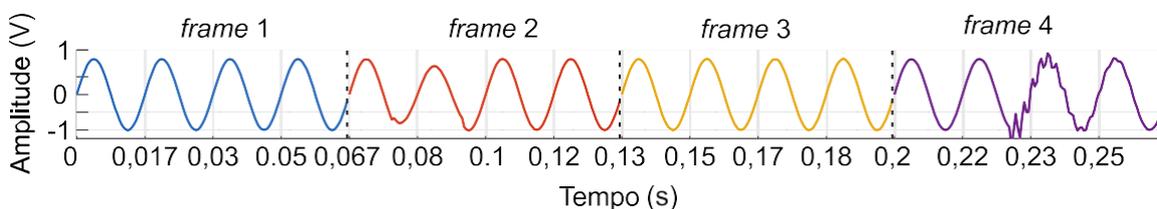
Através da saída *out_4*, é possível ver o início e fim de cada etapa do processamento. O intervalo (a) corresponde ao tempo de inicialização do processador compreendendo a “instalação da matriz \mathbf{G}_0 e vetor de limiares”. Esta etapa deve ser realizada uma única vez no início do processamento, uma vez que esses vetores são fixos. O intervalo (b) corresponde ao tempo para ser efetuada a FFT do sinal. O intervalo (c) corresponde ao tempo para ser efetuada a escolha dos *voices* de maior módulo (não necessário na FST completa, mas necessário na versão com seleção de *voices*). O intervalo (d) corresponde ao tempo necessário para que ocorra a multiplicação de todas as linhas (na FST completa) ou apenas as escolhidas dos *voices* selecionados da matriz \mathbf{G}_0 com a \mathbf{H} e a aplicação da IFFT de cada linha escolhida (na versão com seleção de *voices* são apenas 4 linhas, correspondentes aos *voices* selecionados). O intervalo (e) corresponde ao tempo necessário para calcular um único *voice*, (o intervalo (d) é composto de 4 intervalos (e) na versão com seleção de *voices*). O intervalo (f) corresponde ao tempo necessário para calcular a variância de todas as linhas (ou apenas das selecionadas) e comparar com o limiar. O intervalo (g) corresponde ao tempo necessário para calcular a variância de uma única linha e comparar com o limiar específico dessa linha (o intervalo (f) é composto por 4 intervalos (g) na versão com seleção de *voices*).

4.2 Aprimoramento da FST com seleção de *voices*

Para a segunda versão, foi utilizado um sinal contendo 512 pontos com 4 *frames*, conforme mostrado na Figura 10.

O sinal teve um ruído adicionado com SNR = 45 dB aos quatro *frames*. O

Figura 10 – Sinal para a versão com seleção de *voices*.



Fonte: Elaborado pelos próprios autores, 2024.

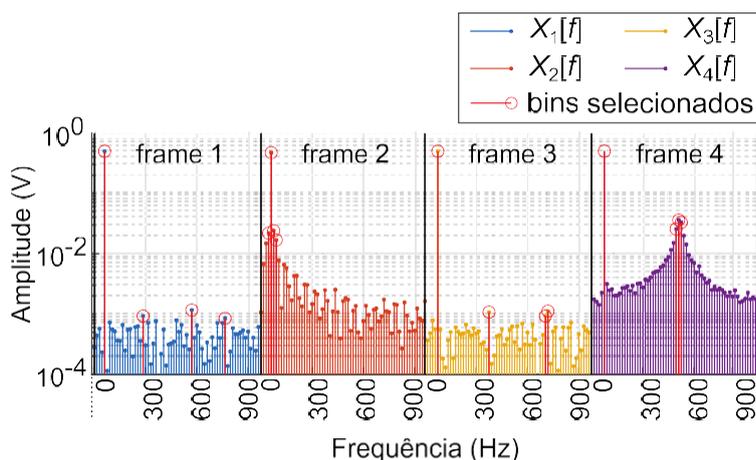
primeiro *frame* possui apenas a frequência fundamental, portanto deve ser classificado como um *frame* de não novidade. O segundo *frame* possui um distúrbio *sag* durante o segundo ciclo, portanto deve ser um *frame* de novidade. O terceiro *frame* volta a ter apenas a frequência fundamental, ou seja, não contém novidades. O quarto e último *frame* possui um transiente começando no terceiro ciclo, sendo um *frame* de novidade.

O processo de seleção de *voices* pode ser visto na Figura 11, onde, a partir da FFT, foram selecionados os 4 bins de maior energia para cada *frame*.

A Figura 12 compara os resultados da variância dos *voices* selecionados para cada *frame* obtidos da simulação funcional de *hardware* em relação aos resultados do MATLAB®. Pelos resultados do simulador, podem ser observados os bins selecionados e se foram detectadas novidades em cada um deles. Foram indicadas novidades no segundo e quarto *frames*, conforme esperado pelos resultados obtidos no MATLAB®.

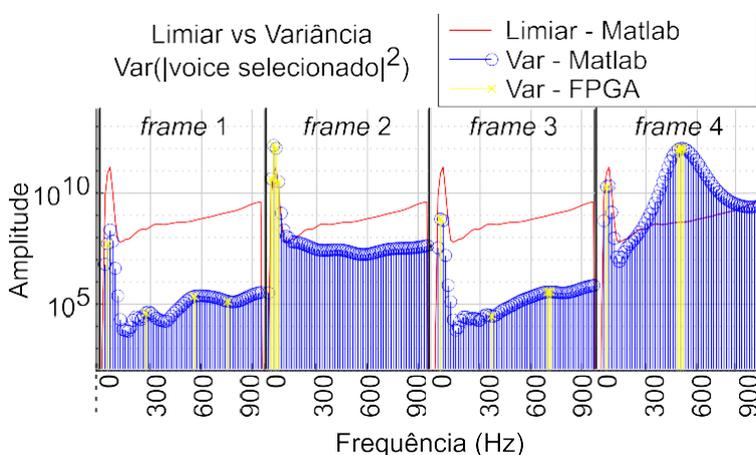
A Figura 13 mostra o diagrama de temporização para a implementação da versão com seleção de *voices*. As mesmas nomenclaturas dos intervalos vistas na Figura 9 foram utilizadas.

Figura 11 – Transformada de Fourier de cada *frame* e bins selecionados.



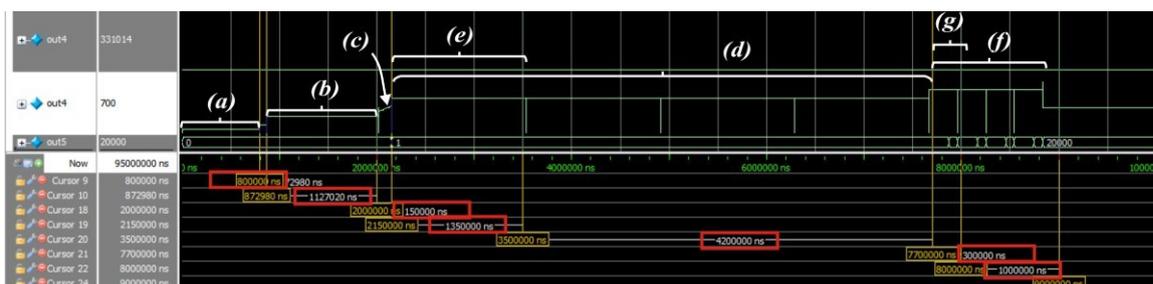
Fonte: Elaborado pelos próprios autores, 2024.

Figura 12 – Aplicação dos limiares às variâncias dos *voices* selecionados de cada *frame*.



Fonte: Elaborado pelos próprios autores, 2024.

Figura 13 – Análise temporal da 2ª simulação. Um único *frame*.



Fonte: Elaborado pelos próprios autores, 2024.

As durações dos intervalos de processamento das duas implementações foram organizadas na Tabela 1.

Pode-se ver que houve uma diminuição significativa de 109 ms da FST completa para 8,9 ms da versão com seleção de *voices*. Uma outra observação importante é que a versão completa não seria possível de ser calculada antes do

término de um *frame* com um *clock* de 50 MHz, ao passo que a estratégia de seleção de *voices* possibilitou essa execução.

Tabela 1 – Comparação da duração da implementação da FST completa com a estratégia de seleção de *voices*

FST completa		Seleção de <i>voices</i>	
(a)	800 μ s	(a)	800 μ s
(b)	1,1 ms	(b)	1,1 ms
(c)	150 μ s	(c)	150 μ s
(d)	8 ms	(d)	5,6 ms
(e)	140 μ s	(e)	140 μ s
(f)	19 ms	(f)	1,3 ms
(g)	300 μ s	(g)	300 μ s

Fonte: Elaborado pelos próprios autores, 2024.

5 CONCLUSÃO

O presente trabalho apresentou a implementação em FPGA da ST com aplicação em detecção de novidades em sinais QEE. Uma formulação da ST adaptada para ser executada em um processador customizado em ponto flutuante foi descrita, denominada FST. Duas versões implementadas foram apresentadas. A primeira, que calcula a FST completa considerando todos os *voices* e a segunda, utilizando a estratégia de seleção de *voices*. Os resultados mostraram que a ST é uma excelente ferramenta para a detecção de quebras de estacionariedade dentro dos *frames* e que a estratégia de seleção de *voices* foi eficaz na busca pela redução do tempo de processamento, ao mesmo tempo que preservou as informações de variação no tempo do sinal analisado. A seleção de *voices* reduziu o tempo total de processamento de um *frame* de 114 ms para 8,2 ms, o que corresponde a 92,8% de redução, permitindo que o processamento de um *frame* de 128 pontos amostrados a uma taxa de $F_s = 1920$ Hz pudesse ser concluído dentro do período de um *frame*, ou seja, 66,6 ms, com um *clock* de 50 MHz presente em placas de FPGA de baixo custo. Como trabalhos futuros, propõe-se a validação do detector em bancada através da implementação em uma placa FPGA e testes com sinais reais medidos em campo.

AGRADECIMENTOS

Os autores agradecem à UFJF, à Pró-Reitoria de Pós-Graduação e Pesquisa (PROPP) da UFJF, à CAPES e ao CNPQ pelo apoio na realização do

trabalho. Agradecemos também o apoio dos membros do laboratório NIPS/UFJF.

REFERÊNCIAS

BALASUBBAREDDY, M.; SRI RAM, Kondapalli Venkata; SANGU, Ravindra. Modeling and design of FPGA-based power quality analyzer.

In: **Microelectronics, Circuits and Systems: Select Proceedings of Micro2021**. Singapore: Springer Nature Singapore, 2023. p. 433-443. Doi: https://doi.org/10.1007/978-981-99-0412-9_39.

DE OLIVEIRA, Roger Alves; BOLLEN, Math HJ. Deep learning for power quality. **Electric Power Systems Research**, v. 214, p. 108887, 2023. Doi: <https://doi.org/10.1016/j.epsr.2022.108887>.

KAPISCH, Eder Barboza. **Estimação de frequência e detecção de novidades aplicadas à compressão de sinais elétricos de potência**: Tese (Doutorado em Engenharia Elétrica) – Universidade Federal de Juiz de Fora, Juiz de Fora, 2019. <https://drive.google.com/file/d/1qYnJ-KNjwaZM9-8SQ7k-9vWqSnHssgLK/view?usp=sharing>.

KAPISCH, Eder B. *et al.* Spectral variation-based signal compression technique for gapless power quality waveform recording in smart grids. **IEEE Transactions on Industrial Informatics**, v. 18, n. 7, p. 4488-4498, 2021. Doi: <https://doi.org/10.1109/TII.2021.3121295>.

KATARAY, Tarun *et al.* Integration of smart grid with renewable energy sources: Opportunities and challenges—A comprehensive review. **Sustainable Energy Technologies and Assessments**, v. 58, p. 103363, 2023. Doi: <https://doi.org/10.1016/j.seta.2023.103363>.

KUMAR, Rajat *et al.* Online identification of underlying causes for multiple and multi-stage power quality disturbances using S-transform. **IETE Journal of Research**, v. 69, n. 6, p. 3739-3749, 2023. Doi: <https://doi.org/10.1080/03772063.2021.1913073>

PRESS, William H. *et al.* Numerical recipes in C++: The art of scientific computing (2nd edn) 1 numerical recipes example book (C++)(2nd edn) 2 numerical recipes multi-language code CD ROM with LINUX or UNIX single-screen license revised version3. **European Journal of Physics**, v. 24, n. 3, p. 329-330, 2003. Doi: <https://doi.org/10.1088/0143-0807/24/3/701>.

RIBEIRO, Victor Mendes *et al.* Real-Time Implementation of Stockwell Transform in FPGA Platform Using Soft-core Processor Applied to Novelty Detection in Power Quality Signals. **Journal of Control, Automation and Electrical Systems**, p. 1-13, 2024. Doi: <https://doi.org/10.1007/s40313-024-01083-z>.

SHANG-BIN, Jiao *et al.* Fast S-transform for fault line selection in distribution network system. In: **2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)**. IEEE, 2018. p. 2303-2308. Doi: <https://doi.org/10.1109/ICIEA.2018.8398094>.

STOCKWELL, Robert Glenn; MANSINHA, Lalu; LOWE, R. P. Localization of the complex spectrum: the S transform. **IEEE transactions on signal processing**, v. 44, n. 4, p. 998-1001, 1996. Doi: <https://doi.org/10.1109/78.492555>.

PRINCIPIA

CAMINHOS DA INICIAÇÃO CIENTÍFICA
UNIVERSIDADE FEDERAL DE JUÍZ DE FORA
PRÓ-REITORIA DE POS-GRADUAÇÃO E PESQUISA (PROPP)
UFJF.BR/PROPP