

Implementação da Transformada de Stockwell em Plataforma FPGA Aplicada à Detecção de Novidades em Sinais de Qualidade de Energia Utilizando Processador *Soft-core*

Victor Mendes Ribeiro * Naiara da Silva Maia dos Santos * Eder Barboza Kapisch *
Leandro Rodrigues Manso Silva * Carlos Augusto Duque *

* Departamento de Engenharia Elétrica, Universidade Federal de Juiz de Fora, MG,
(e-mail: victormrmat@gmail.com).

Abstract: Considering the smart grids (SGs) establishment scenario, where the presence of non-linear loads and new power generation forms become increasingly expressive, there is a potential for the emergence of new disturbances. Furthermore, considering the huge amount of data coming from smart meters, it is necessary to use novelty detection methods in voltage and current waveform signals, in order to preserve the relevant information and promote efficient storage of the data. The Stockwell Transform (ST) is a time-frequency distribution that has shown great ability to detect novelties related to stationarity changes in signals. Therefore, the present work describes the use of ST for the purpose of detecting novelties in Power Quality (PQ) signals. The implementation of this transform in FPGA is proposed through the use of a Soft-Core processor to optimize the hardware resources of the FPGA. In addition, a voice selection strategy is proposed in order to reduce the complexity of the algorithm, as well as reduce the processing time, while maintaining the detection capability.

Resumo: Diante do cenário de consolidação das Redes Elétricas Inteligentes (REIs), onde a presença de cargas não lineares e novas formas de geração de energia elétrica se tornam cada vez mais expressivas, existe um grande potencial para o aparecimento de novos distúrbios. 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 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 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 *voices*, a fim de diminuir a complexidade do algoritmo, bem como diminuir o tempo de processamento, mantendo-se a capacidade de detecção.

Keywords: Stockwell Transform. FPGA. Embedded processor. Power Quality.

Palavras-chaves: Transformada de Stockwell. FPGA. Processador embarcado. Qualidade de energia.

1. INTRODUÇÃO

O crescimento dos Sistemas Elétricos de Potência (SEP) e a inserção de novos tipos de cargas não lineares à rede elétrica, a tornou cada vez mais ramificada, complexa e suscetível a novos tipos de distúrbios elétricos, observáveis através das formas de onda de corrente e tensão. Com os avanços tecnológicos da informática e do poder de processamento dos dispositivos de medição e controle presentes no SEP, tem-se observado a implementação das Redes Elétricas Inteligentes (REI), conhecidas também como *Smart Grids* (SG).

As REI buscam atender a nova demanda de gerenciamento de recursos elétricos, a fim de que o sistema possa operar de forma autônoma, sendo capaz de analisar os defeitos e falhas, procurando mitigar os problemas de Qualidade de Energia Elétrica (QEE) da rede em tempo real. Para que a REI possa ser capaz de identificar os possíveis novos

tipos de distúrbios elétricos ainda não conhecidos, todas as informações relevantes presentes nos sinais de forma de onda de tensão e corrente precisam ser preservadas. Essa tarefa se torna desafiadora quando consideramos a quantidade de equipamentos captando dados de medição em tempo real.

Quanto maior o número de dispositivos inteligentes registradores das grandezas elétricas, maior a quantidade de dados a serem armazenados e analisados, termo conhecido como *big data* (Hongxun et al., 2018). Desta forma, se faz necessária a utilização de técnicas de processamento de sinais que permitam um armazenamento eficiente desses dados a fim de diminuir a informação trafegada pela rede (Kapisch et al., 2021), e, ao mesmo tempo, sejam capazes de preservar as informações relevantes do sinal analisado. É nesse cenário que as técnicas de detecção de novidades podem cumprir um papel importante (Rodrigues Manso Silva et al., 2016).

Algumas técnicas têm se mostrado promissoras em relação à detecção de novidades, dentre elas, destaca-se a Transformada de Stockwell (*Stockwell Transform*) (ST), com interessantes propriedades teórico-matemáticas e alta capacidade de detecção de quebras de estacionariedade (Stockwell et al., 1996). Sua presença na literatura tem sido cada vez mais frequente. A seguir, alguns trabalhos correlatos

Em (Dash et al., 2003), a ST discreta é mostrada como uma poderosa ferramenta de análise para detecção, localização e classificação de problemas de QEE, em contraponto a Transformada Wavelet (WT). Em (Leonowicz et al., 2009), a ST é utilizada para rastrear as mudanças na amplitude e frequência de sinais elétricos não estacionários. Sinais contendo eventos típicos do SEP, como transitórios de alta frequência provenientes de chaveamento de banco de capacitores, são utilizados para testes. Em (Ferreira et al., 2009), é apresentada a utilização da ST para detectar e classificar diversos distúrbios de QEE. Eventos como transitórios impulsivos são utilizados para mostrar a superioridade da ST sobre a transformada *wavelet* tradicional.

Em (Naik and Kundu, 2011), é mostrado que a aplicação da ST oferece distintas vantagens sobre técnicas como Transformada Wavelet Contínua (CWT) e Transformada de Fourier de Tempo Curto (*Short-time Fourier Transform*) (STFT), por sua correção de fase e melhor localização na frequência mesmo em cenários ruidosos. Em (Pujiantara et al., 2016), uma modificação sobre a ST é proposta, aumentando o controle da janela gaussiana presente na transformada. Informações de ambos os domínios tempo e frequência são extraídas da ST modificada. A referência (Shang-bin et al., 2018) propõe a utilização de uma versão para o cálculo rápido da ST, a qual reduz a complexidade da versão original da ST, ao mesmo tempo que mantém as características tempo-frequência de interesse presentes na transformada.

Em (Kankale et al., 2021), apresenta a classificação de distúrbios de QEE em SEP utilizando a ST juntamente com máquina de vetor de suporte (SVM). É proposto um algoritmo classificador baseado em um limiar de energia a partir da ST. São analisados distúrbios do tipo *sag*, *swell* e interrupção. Por fim, o trabalho descrito em (Kapisch, 2019), elenca diversos tipos de técnicas de detecção de novidades e faz comparação entre elas, mostrando que a ST possui melhor desempenho dentre as técnicas comparadas.

Com isto em mente, a motivação para a implementação da ST proposta neste trabalho vem de suas propriedades de interesse para detecção de novidades em sinais variantes no tempo, muitas delas exploradas nos trabalhos citados. Além disso, o presente trabalho descreve a utilização da *Field-Programmable Gate Array* (matriz de portas programáveis) (FPGA) como plataforma para a implementação da ST para detecção de novidades em tempo real. Tal plataforma é frequentemente utilizada devido à sua flexibilidade na modelagem dos circuitos digitais e poder de processamento paralelo que ela proporciona.

Ainda, é possível utilizar processadores *soft-cores* customizáveis dentro da FPGA, buscando a otimização dos recursos de *hardware* disponíveis. Sendo assim, o principal objetivo deste trabalho constitui-se em elaborar a implementação em *hardware* de um detector de novidades

baseado na ST, no desenvolvimento de aprimoramentos através da seleção de *voices* da transformada e na utilização do *soft-core Scalable Architecture Processor for Hardware Optimization* (SAPHO) (Luiz et al., 2022), para desenvolver a implementação da ST em FPGA, bem como validar sua verificação através de simulação funcional da implementação.

Para a geração dos sinais sintéticos de QEE e simulações do algoritmo e dos aprimoramentos propostos na ST, foi utilizado o ambiente MATrix LABoratory (MATLAB[®]). Para a descrição do *hardware* e síntese dos blocos na FPGA foi utilizado o sintetizador Quartus Prime[®], da Fabricante Intel[®]. A linguagem utilizada foi verilog. Para a simulação funcional do *hardware* e dos aprimoramentos propostos de seleção de *voices*, foi utilizado o *software* de simulação de *hardware* ModelSim[®].

2. REVISÃO TEÓRICA

A ST é uma distribuição tempo-frequência que pode ser utilizada para a análise de sinais variantes no tempo. Ela é uma transformada 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 $e^{-i2\pi ft}$ 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.

A ST é uma extensão das ideias da CWT e é baseada em uma janela gaussiana deslizante e escalável aplicada a bases senoidais fixas no tempo. Ela possui algumas características interessantes que não estão presentes na CWT.

A ST é uma ferramenta que provê uma resolução variável e dependente da frequência ao mesmo tempo que mantém uma direta relação com o espectro de Fourier através da média temporal. Essas vantagens presentes na ST são devidas ao fato de que as ondas senoidais moduladas que servem como base para a transformação não são móveis em relação ao eixo dos tempos, mas fixas, enquanto a janela gaussiana (que serve como um envelope) é dilatada e transladada (Kapisch, 2019).

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, e a transformada de Fourier, calculada pela Transformada Rápida Fourier (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).

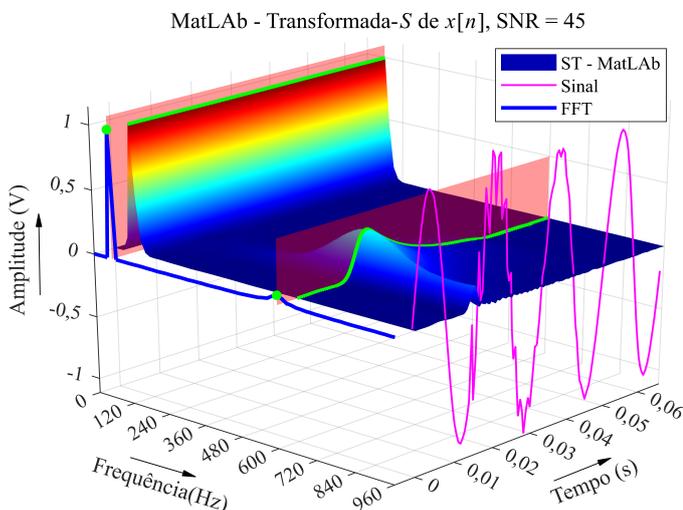


Figura 1. Superfície da Transformada S.

O método de detecção de novidades apresentado aqui utiliza a variância dos *voices* extraídos da superfície do módulo da ST. Entretanto, calcular todas as variâncias de todos os *voices* é uma tarefa computacional que exige uma grande quantidade de processamento. Pensando nisso, em (Kapisch, 2019), foi proposta uma estratégia de seleção de *voices*, que tira proveito das propriedades de recuperação da FFT a partir da Transformada de Fourier (FT), as quais permitem 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. IMPLEMENTAÇÃO EM FPGA DO DETECTOR DE NOVIDADES

Para a implementação de um detector de novidade em FPGA utilizando a teoria da variância dos *voices* da ST, se faz necessária a criação de um algoritmo capaz de efetuar operações sequenciadas com números complexos como FFT, a Transformada Rápida Inversa Fourier (IFFT),

média e variância. Tais operações seriam muito complexas e demandariam muitos recursos de *hardware* caso fossem descritas diretamente em linguagem de descrição de *hardware*. Para contornar este problema, optou-se por utilizar um processador *soft-core* embarcado na FPGA.

3.1 Processador embarcado - SAPHO

O processador *soft-core*, através de seu sequenciamento de comandos, torna as operações necessárias mais próximas aos algoritmos utilizados na criação da teoria proposta. Foi utilizado o SAPHO, 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. Foi criado também um bloco “berço” a fim de realizar a interface das saídas e entradas do processador.

A Figura 2 mostra o processador embarcado compilado pela Integrated Development Environment ou Ambiente de Desenvolvimento Integrado (IDE) do SAPHO, bem como os blocos que formam o “berço” para receber o processador na FPGA. É possível notar que existem 6 registradores de saída na parte direita e um multiplexador com duas entradas na parte esquerda. A 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, etc., conforme mostrado num recorte das diretivas de compilação ao centro da Figura 2.

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 da plataforma IDE SAPHO que são convertidos em instruções em *Assembly* pelo compilador C e em seguida convertidas em código de máquina pelo compilador *Assembler*, ambos pertencentes à IDE. A linguagem utilizada para programar o processador SAPHO é formada por um subconjunto da linguagem de programação C, batizada de C+-. Sendo assim, a implementação do detector de novidades proposto neste trabalho fica limitada a um algoritmo descrito em C+-.

3.2 Implementação do detector em C+-

A implementação do detector pode ser separada em blocos como mostrado na Figura 3. Primeiramente, o sinal é dividido em *frames* de tamanho fixo contendo N_c 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.

Segmentação: Os *frames* possuem tamanho fixo com comprimento de N amostras e não possuem sobreposição.

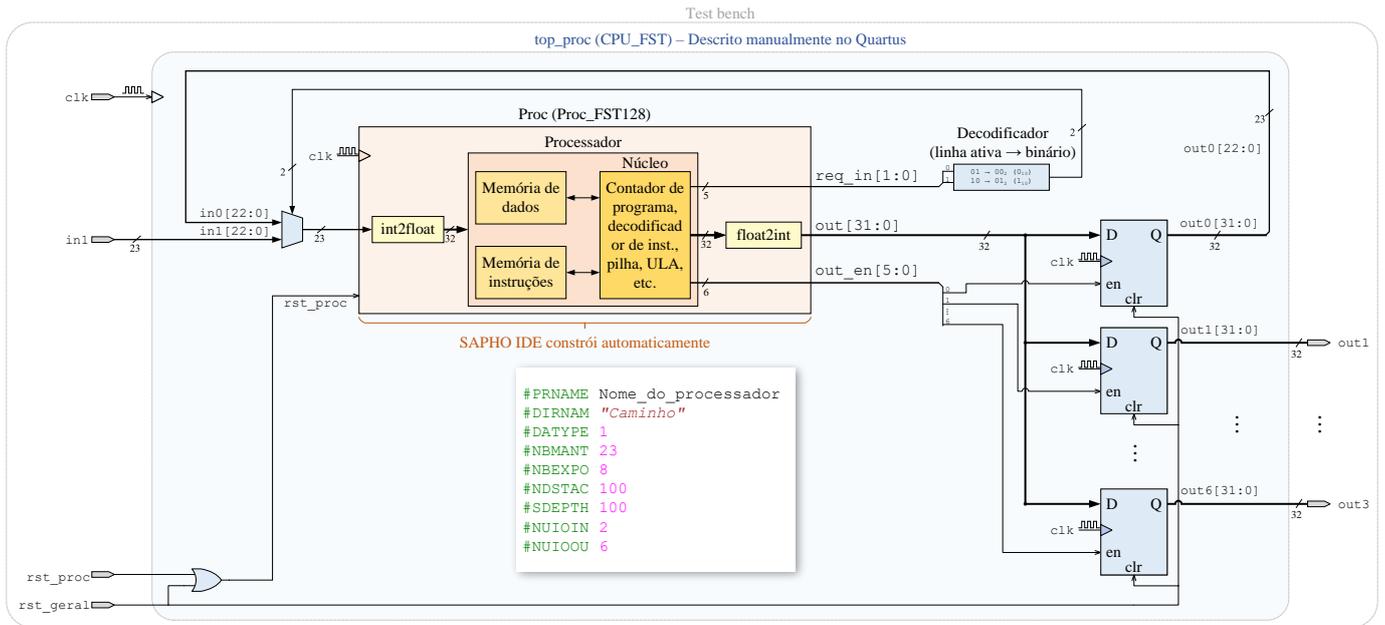


Figura 2. Diagrama do hardware implementado.

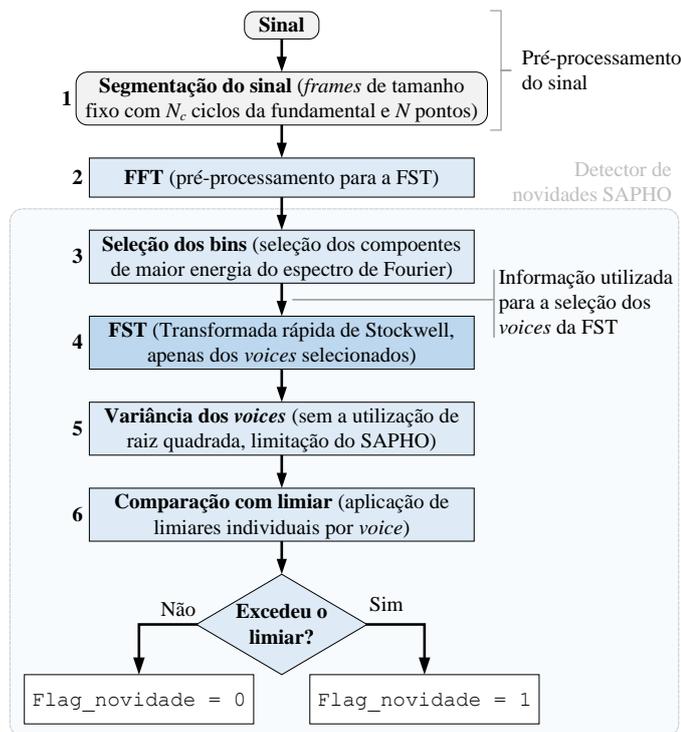


Figura 3. Diagrama do algoritmo de detecção.

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 $F_s = f_1 \times \frac{N}{N_c} = f_0 \times \frac{N}{4}$. A frequência de amostragem adotada foi de 1920 Hz, ou 32 pontos por ciclo da fundamental, resultando em $N = 128$ pontos por *frame* analisado. A Figura 4 mostra um exemplo de sinal segmentado conforme descrito acima.

Transformada de Rápida de Fourier (FFT): Para construir o código da função FFT e IFFT, foi utilizado o código em C++ baseado na (Press et al., 2003), onde foram necessárias apenas algumas modificações para atender às peculiaridades da linguagem C++ do SAPHO. Nesse có-

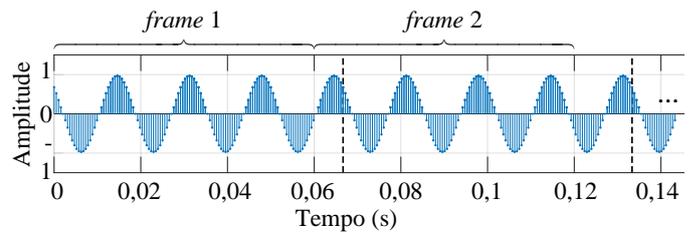


Figura 4. Separação do sinal em frames.

modo, os parâmetros de entrada e saída da transformada são criados, de forma que a ordem dos índices do eixo de frequências são idênticos à do MATLAB®, conforme a Figura 5.

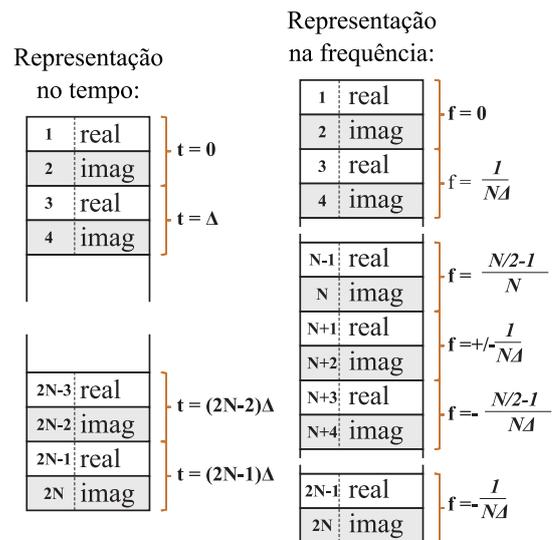


Figura 5. Alocação dos dados complexos em um array.

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 vozes 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.

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

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

em que $\mathbf{S}\left(mT, \frac{n}{NT}\right)$ 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 $T = \frac{1}{F_s}$ é o período de amostragem. $\mathcal{F}^{-1}\{\cdot\}$ representa a transformada inversa de Fourier, $X(\cdot)$ é a FFT de $x(t)$ e \odot 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:

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

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 da \mathbf{G}_0 no processador sem a necessidade de ser calculada mais de uma vez. Para isso, foi criada uma função em MATLAB[®] que escreve um arquivo de texto .txt em linguagem C++, contendo os comandos de alocação dos valores de \mathbf{G}_0 em um *array*, onde estes comandos serão executados junto às instruções de programa no SAPHO.

Desta forma, para a implementação da FST, basta a criação de um função que, ao inicializar o processador, os valores da matriz \mathbf{G}_0 sejam alocados 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, conforme mostrado na Figura 6.

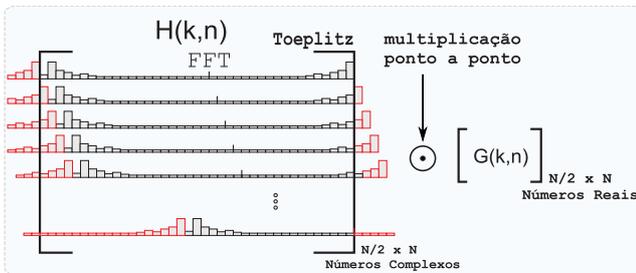


Figura 6. Diagrama da criação da função FST.

Cálculo da variância dos voices: Devido a 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).

$$Var_n = \sum_{n=1}^N \frac{(|\mathbf{v}_n[m]|^2 - \overline{|\mathbf{v}_n[m]|^2})^2}{N - 1}, \quad (4)$$

onde Var_n é a variância do módulo ao quadrado do vetor que representa o *voice* $\mathbf{v}_n[n]$ e $\overline{|\mathbf{v}_n[m]|^2}$ é o valor médio de \mathbf{v}_n .

Criação dos limiares: Para criar um detector factível para implementação, é necessário criar um limiar como uma certa flexibilidade a fim de evitar ocorrências de eventos falsos positivos na detecção. Para isso, foram criados no MATLAB[®] 100 *frames* de sinal com amplitude unitária 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 7, garantindo que 99,9% das possíveis variâncias dos *frames* corrompidos estejam abaixo do limiar.

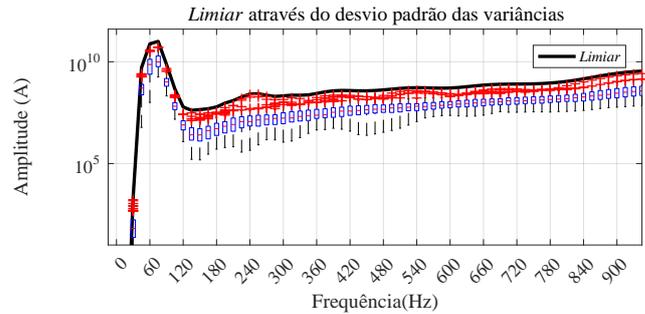


Figura 7. Criação do limiar através desvio padrão.

4. RESULTADOS

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*. O 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 surge um transitório com frequência 600 Hz que decai exponencialmente, conforme mostrado na Figura 8.

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 9. Pode-se ver que os resultados obtidos pelo processador embarcado na FPGA correspondem aos esperados.

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 10.

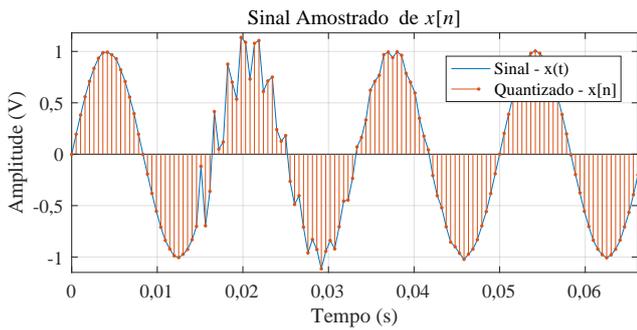


Figura 8. *Frame* de sinal utilizado na simulação.

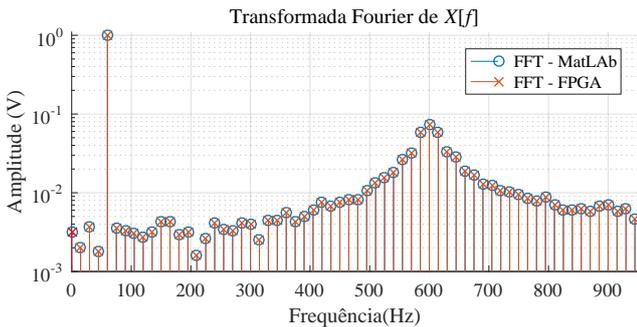


Figura 9. Cálculo da FFT.

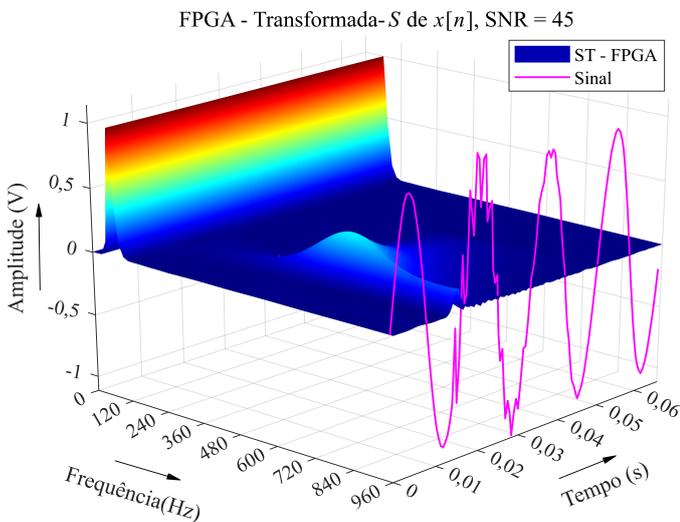


Figura 10. Comparação dos resultado da FST.

O *frame* de sinal aqui analisado foi propositadamente escolhido como o mesmo utilizado no exemplo mostrado na Figura 1. 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 11. 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.

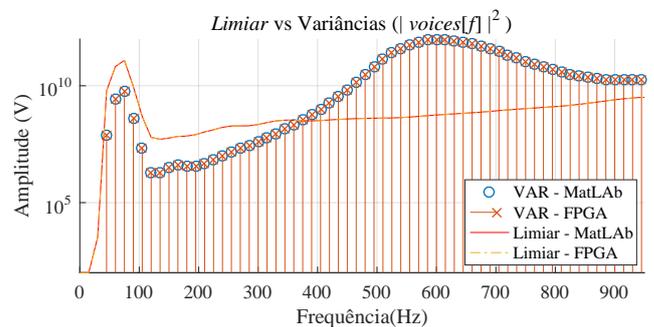


Figura 11. Criação do limiar através desvio padrão.

Na simulação funcional, foi utilizado um *clock* de 50 MHz e um *frame* contendo $N = 128$ pontos. A Figura 12 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

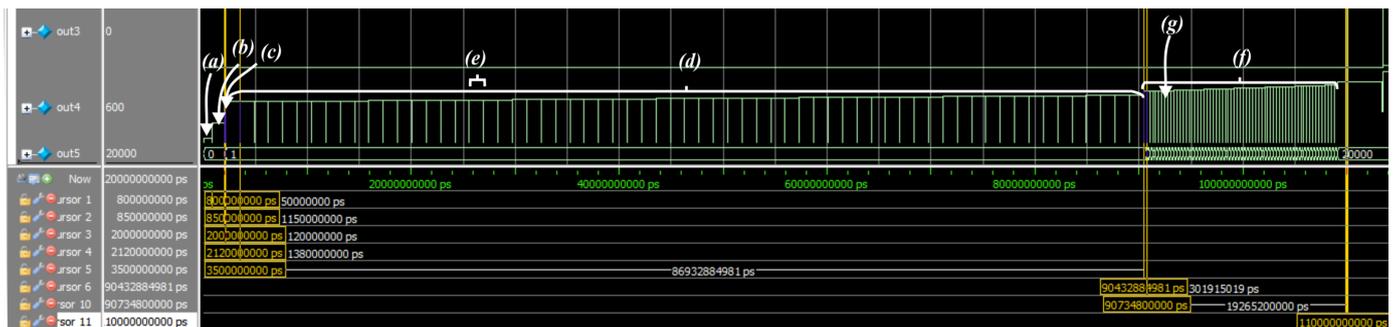


Figura 12. Análise temporal da 1ª simulação.

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 FST com seleção de *voices*

Para a segunda versão, foi utilizado um sinal contendo 512 pontos com 4 *frames*, conforme mostrado na Figura 13.

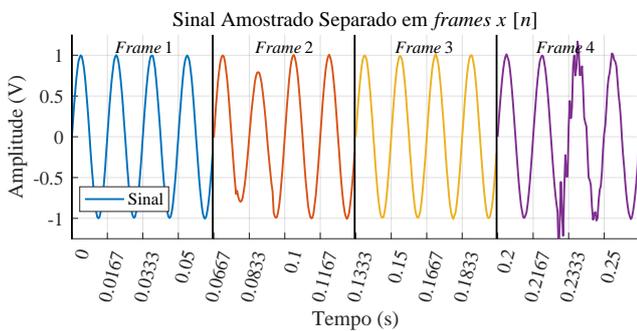


Figura 13. Sinal para a versão com seleção de *voices*.

O sinal teve um ruído adicionado com SNR = 45 dB aos quatro *frames*. O 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 transitório começando no terceiro ciclo, sendo um *frame* de novidade.

O processo de seleção de *voices* pode ser visto na Figura 14 onde, a partir da FFT, foram selecionados os 4 bins de maior energia para cada *frame*.

A Figura 15 compara os resultados da variância do *voices* selecionados para cada *frame* obtidos da simulação funcional de *hardware* em relação aos resultados do MATLAB®. Pelos resultados do simulador, pode ser observado os *bins* selecionados e se foram detectadas novidades em cada um deles. Foram indicadas novidades no segundo e quarto

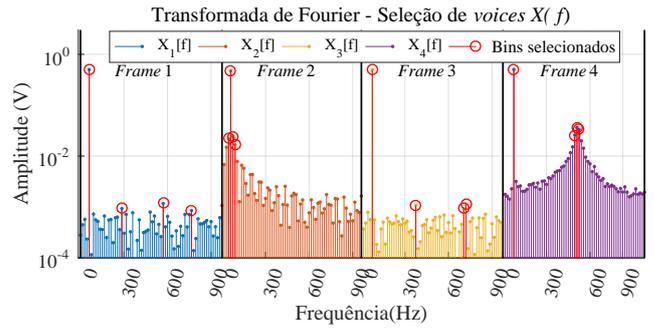


Figura 14. Transformada de Fourier de cada *frame* e bins selecionados.

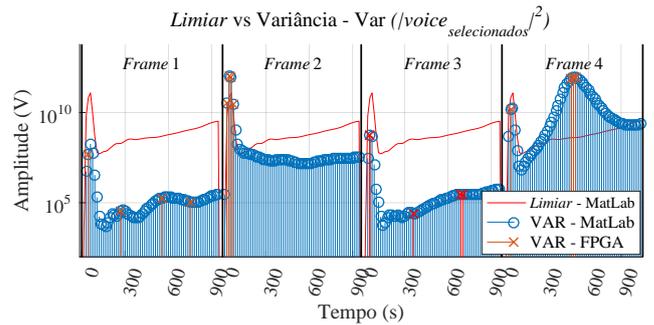


Figura 15. Aplicação dos limiares à variâncias dos *voices* selecionados de cada *frame*.

frames, conforme esperado pelos resultados obtidos no MATLAB®.

A Figura 16 mostra o diagrama de temporização para a implementação da versão com seleção de *voices*. As mesmas nomenclaturas dos intervalos vistas na Figura 12 foram utilizadas.

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

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 µs	a) 800 µs
b) 1.1 ms	b) 1.1 ms
c) 150 µs	c) 150 µs
d) 8 ms	d) 5.6 ms
e) 140 µs	e) 140 µs
f) 19 ms	f) 1.3 ms
g) 300 µs	g) 300 µs
Total 114 ms	Total 8.2 ms

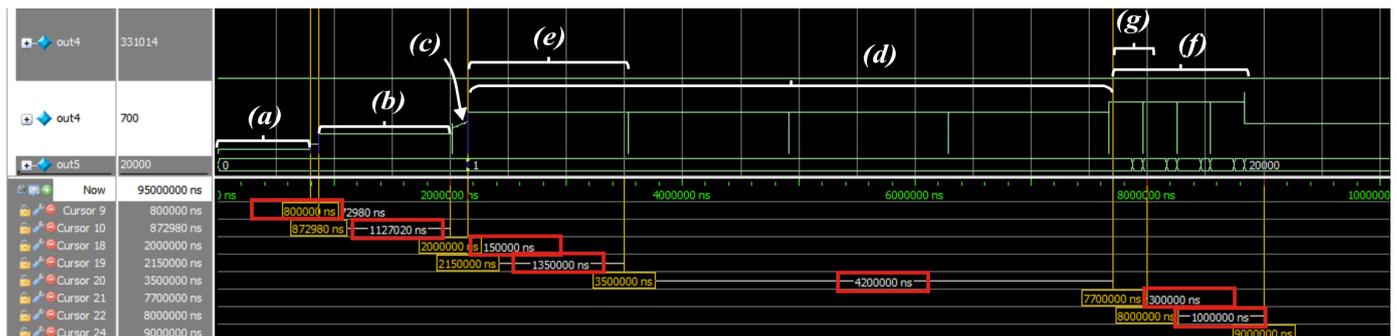


Figura 16. Análise temporal da 2ª simulação. Um único *frame*.

Pode-se ver que houve uma diminuição significativa de 114 ms da FST completa para 8,2 ms da a 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.

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.

6. AGRADECIMENTOS

Os autores agradecem ao CNPq, CAPES e INERGE que financiaram este trabalho.

REFERÊNCIAS

- Dash, P., Panigrahi, B., and Panda, G. (2003). Power quality analysis using s-transform. *IEEE Transactions on Power Delivery*, 18(2), 406–411. doi:10.1109/TPWRD.2003.809616.
- Ferreira, D.D., Marques, C.A.G., Cerqueira, A.S., Duque, C.A., and Ribeiro, M.V. (2009). Sistema automático de detecção e classificação de distúrbios elétricos em qualidade da energia elétrica. *Sba: Controle & Automação Sociedade Brasileira de Automática*, 20(1), 53–62.
- Hongxun, T., Honggang, W., Kun, Z., Mingtai, S., Haosong, L., Zhongping, X., Taifeng, K., Jin, L., and Yaqi, C. (2018). Data quality assessment for on-line monitoring and measuring system of power quality based on big data and data provenance theory. In *2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, 248–252. doi:10.1109/ICCCBDA.2018.8386521.
- Kankale, R., Paraskar, S., and Jadhao, S. (2021). Classification of power quality disturbances in emerging power system using s-transform and support vector machine. In *2021 IEEE 2nd International Conference On Electrical Power and Energy Systems (ICEPES)*, 1–6. doi:10.1109/ICEPES52894.2021.9699673.
- Kapisch, E.B. (2019). Estimação de frequência e detecção de novidades aplicadas à compressão de sinais elétricos de potência / eder barboza kapisch.. In *Tese (Doutorado) – Universidade Federal de Juiz de Fora, Faculdade de Engenharia. Programa de Pós-Graduação em Engenharia Elétrica*, 2019.
- Kapisch, E.B., De Morais, V.V., Silva, L.M., Manhaes, L., and Duque, C. (2021). Spectral variation-based signal compression technique for gapless power quality waveform recording in smart grids. *IEEE Transactions on Industrial Informatics*, 1–1. doi:10.1109/TII.2021.3121295.
- Leonowicz, Z., Lobos, T., and Wozniak, K. (2009). Analysis of non-stationary electric signals using the s-transform. *COMPEL-The international journal for computation and mathematics in electrical and electro-nic engineering*.
- Luiz, M.M., Duque, T.F., De Almeida, A.H.S., Kapisch, E.B., Silva, L.R.M., and Lima, M.A.A. (2022). Power quality parameters calculation using fpga embedded parallel processors in compliance with the iec 61000-4-30 standard. *Journal of Control, Automation and Electrical Systems*, 1–12. doi:10.48011/asba.v2i1.1287.
- Naik, C.A. and Kundu, P. (2011). Identification of short duration power quality disturbances employing s-transform. In *2011 International Conference on Power and Energy Systems*, 1–5. doi:10.1109/ICPES.2011.6156671.
- Press, W.H., Teukolsky, S.A., Vetterling, W.T., and Flannery, B.P. (2003). 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.
- Pujiantara, A.E.M., Priyadi, A., Pujiantara, M., Penangsang, O., Anggriawan, D.O., and Tjahjono, A. (2016). Improvement of power quality monitoring based on modified s-transform. In *2016 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, 539–544. doi:10.1109/ISITIA.2016.7828717.
- Rodrigues Manso Silva, L., Barbosa Kapish, E., Augusto Duque, C., Manhes de Andrade Filho, L., Santiago Cerqueira, A., and Fernando Ribeiro, P. (2016). The concept of novelty detection applied to power quality. In *2016 IEEE Power and Energy Society General Meeting (PESGM)*, 1–5. doi:10.1109/PESGM.2016.7741689.
- Shang-bin, J., Yun-peng, Y., Yan-jin, S., Wen-wen, S., and Jing-wen, G. (2018). Fast s-transform for fault line selection in distribution network system. In *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 2303–2308. doi:10.1109/ICIEA.2018.8398094.
- Stockwell, R., Mansinha, L., and Lowe, R. (1996). Localization of the complex spectrum: the s transform. *IEEE Transactions on Signal Processing*, 44(4), 998–1001. doi:10.1109/78.492555.