Introduction and evaluation of a new time-frequency transformation based on the CQT

Prof. Dr. Lilia, Lajmi

Kai Blum, B.Eng.
Marc-André Tucholke, Student
University of applied Sciences, Wolfenbüttel, Germany

DOI: 10.26271/opus-1254

In this paper a new time-frequency transform based on the constant-Q transform, named as adaptive quality frequency transform (AQFT), is proposed. The AQFT is a non-uniform transform with logarithmically spaced center frequencies. The CQT offers a computational disadvantage due to the large number of samples needed to assure the quality and the desired resolution at the lower frequency range. The AQFT addresses this problem precisely and limits the number of samples required by a comparable frequency division. Finally, computer simulations are provided to illustrate the performance of the proposed algorithm. According to the simulation results, the accuracy and the processing time of the AQFT and of traditional time-frequency analysis methods such as the Fast Fourier Transform (FFT), the Signal Derivate FFT (SD-FFT) and the wavelet packet transform (WPT) are contrasted.

0 INTRODUCTION

The sinusoidal transform and time-frequency representation play an increasingly important role in the field of signal processing. The sinusoidal transform provides a way of representing an audio signal as a sum of sinusoids. This procedure is of great importance for some applications in the field of audio coding or modeling. Audio coding for cochlea implants, automatic speech recognition, music information retrieval are examples of current research areas.

Ideally, a sinusoidal transform should represent the sound information in a manner that reflects human perception. In human hearing the frequency scale is compressed in a logarithmic way. At low frequencies about 500 Hz, the relation between physical differences in frequency and perceived differences is roughly linear. Above that point, it is approximately logarithmic [1]. The estimation of human frequency resolution is a result of studies of masking.

Various transforms are currently being used. The best known and most used transform is the fast fourier transform (FFT). It is a known fact that the FFT-based estimation have good accuracy for harmonics if the signal is stationary.

However, the FFT has some serious drawbacks. The analysis results in a linear frequency scale with a constant frequency resolution, which does not reflect human hearing. These differences between the FFT and the psychoacoustic requirements cause e.g. in coding/decoding or transmission quality restrictions. Then there is the loss of time information in transforming to the frequency domain. Using the FFT as time-frequency representation, it is impossible to find out when a particular event has taken place. This information may not be very important for stationary signals. However, for signals with no stationary or transitory characteristics FFT is not suitable.

Enhancements in using FFT under consideration of the time resolution represents the Short Time Fourier Transform (STFT). The STFT analyses a small section of the signal at a time. One particular size of the time window is selected for all the frequencies. The subdivision of the signal in this way essentially ignores the effect of interferers-nearby sinusoids whose sidelobes may tilt magnitude spectrum peaks slightly so that they no longer correspond exactly to sinusoidal components [2]. The mentioned problems of the constant resolution in the FFT spectrum, the poor temporal resolution of the STFT and the associated risk of the peaks overlapping in the STFT spectrum, restrict the flexibility of these transforms. An approach to enhance the accuracy of sinusoidal parameter estimation in STFT spectrum is proposed in [2]. To minimize the problem of the tradeoff of time versus frequency in the FFT, a new method for high precision fourier analysis of sounds using signal derivates (SD-FFT) is proposed in [3] and [4]. This method improves the precision of the fourier analysis not only in frequency and amplitude but also in time resolution. The idea of SD-FFT

thus provides a compromise between time domain and frequency resolution.

When applied to audio or speech signals, these transforms remain unfavorable. The reason for this is, on the one hand, that speech and music signals concentrate most of their energy in the mid-lower part of the spectrum, and therefore ovelaps are more likely to occur in this area. On the other hand, musical notes follow a logarithmic frequency relationship that does not correspond with the linearly spaced subbands of a STFT spectrogram. Notes in the lower range often fall into the same subbands and will thus overlap. [5]

To overcome this, transforms with high frequency resolution for low frequencies and high time resolution for high frequencies are favored. The constant Quality Transform (CQT) proposed by C.J. Brown in 1991 [6] addresses the problem of the constant resolution of the FFT. The goal of the CQT is to transform a signal so that the spectrum has a logarithmic frequency resolution. Unlike the STFT, the CQT provides a varying time-frequency resolution. This results in a high spectral resolution at low frequencies and high temporal resolution at high frequencies [7]. From this point of view, the CQT would be more suitable for music and speech signals.

Computationally, the CQT is expensive as compared with the FFT or the STFT [7]. For performance reasons, Nisar et al proposed in [7] an adaptive method that provides a framework of switching between STFT for narrow band and CQT for wide-band signals, after analyzing the input signal. Numerous authors have contributed to a more efficient CQT calculation [8, 9]. Schörkhuber et al proposed in [9] a computationally inexpensive FFT based CQT analyzer.

Initially, an exact inverse transformation was missing for CQT. In view of the reconstruction methods proposed in [10], [11], [12], CQT has the potential to be more suitable than FFT for applications in music signal processing.

The CQT requires a large number of samples for the analysis in the low frequency range, so that the time resolution is suboptimal. In this paper a modified CQT is proposed with a limitation of the window lengths and an adjustment of the quality for different frequency ranges.

The spectral division of the CQT has a similarity with the wavelet transform. The wavelet transform presents an approach, which satisfies the requirements of more flexibility by varying the window size to determine either time or frequency more accurately [13]. Similar to the CQT, the wavelet analysis allows the use of long time intervals for more precise low-frequency information and shorter time intervals where high-frequency information is required

Wavelet transform is a time-frequency representation of any stationary or non-stationary waveform. It measures similarity between the original waveform and basic function of wavelet transform known as mother wavelet through wavelet coefficients [14]. Wavelet transform can preserve both time and frequency information without any effect on resolution [14]. It has been suggested that wavelets are particularly good at modeling the frequency response of the human auditory system [15].

Wavelet Packet Transform (WPT) is favored over other forms of transform i.e. discrete Wavelet transform (DWT) because it provides uniform frequency bands and offers flexible decomposition through merging splitting process of the nodes. Also it be considered as a generalization of wavelet transform [16].

For evaluation purpose, the performance of the here proposed method and the WPT are compared

This paper is organized as follows: section 1 gives an overview of the theory of CQT and its relationship to FFT and derives the new method, here named the adaptive quality frequency transform (AQFT). The transforms used for the later comparison with the AQFT are explained in section 2 (SDFFT) and 3 (WPT). Finally, we make a comparison of these methods in section 4 using synthetic signals and we discuss the results. Section 5 concludes with a brief summary.

1 THE CONSTANT QUALITY TRANSFORM AND THE ADAPTIVE QUALITY FREQUENCY TRANSFORM

The constant-Q transform is related to the discrete fourier transform and very closely related to the complex Morlet Wavelet transform.

The DFT $X^{dft}(k)$ of a discrete time domain signal $x(n)$ is defined as:

$$X^{dft}(k) = \sum_{n=0}^{N-1} w(n) \cdot x(n) \cdot e^{-j2\pi\frac{nk}{N}} \tag{1}$$

where n and k are the time and frequency parameters respectively and $\{w(n)\}_{n\epsilon[0,N-1]}$ is a normalized analysis window.
N represents the DFT-length.
The spectral resolution Δf of the DFT is defined as follows:

$$\Delta f = \frac{f_s}{N} \tag{2}$$

f_s is the sampling frequency of the signal.

The center frequencies $f_k = k \cdot \Delta f$ with $k \in \{0,1,\cdots,N-1\}$ are distributed uniformly since Δf is constant for all frequencies. So the DFT can be considered as a filter bank with equal spaced center frequencies f_k.

The DFT can also be formulated depending on f_k.

$$X^{dft}(k) = \sum_{n=0}^{N-1} w(n) \cdot x(n) \cdot e^{-j2\pi\frac{n \cdot f_k}{f_s}} \tag{3}$$

If we analyze music signals, it is important to look at the frequency distribution. Musical notes exhibit an exponential frequency distribution $f_k = f_0 \cdot 2^{\frac{k}{b}}$, where f_0 denotes the lower bound of frequencies to be considered.

For $b = 12$ we obtain twelve frequency bins per octave corresponding to the western musical scale of twelve semitones per octave. For b ∈ {24, 36, ...} an even higher resolution than semitone resolution can be achieved which is beneficial if music instruments are not perfectly tuned [10]. There is a similar distribution of the spectral components for speech signals, which could be regarded as a subset of music [17].

For music and speech, a time-frequency representation with geometrically spaced frequency bins, such as the CQT, is more suitable. The CQT can be seen as a filter bank with logarithmically spaced center frequencies f_k. The bandwidth Δ_k^{cq} of the k-th filter is a multiple of the width of the previous filter.

$$\Delta_k^{cq} = \Delta_{k-1}^{cq} \cdot 2^{\frac{1}{b}} \tag{4}$$

b denotes the number of bins (or filters) per octave. b is the most important parameter of choice when using the CQT, because it determines the time-frequency resolution trade-off of the CQT.

The center frequency f_k of the k-th filter can be calculated using the base frequency f_0 (the center frequency of the lowest filter)

$$f_k = f_0 \cdot 2^{\frac{k}{b}}; \quad (k = 0, \cdots, K-1) \tag{5}$$

where K determines the overall number of frequency bins (the total number of filters).

$$K = \left\lceil b \cdot log_2\left(\frac{f_{max}}{f_0}\right)\right\rceil \tag{6}$$

f_{max} is the maximum frequency in the spectrum. f_0 and f_{max} must be specified beforehand.
The factor Q that gives this transform its name represents the quotient between the center frequency and the bandwidth of the filter [18].

$$Q = \frac{f_k}{\Delta_k^{cq}} = \frac{f_k}{f_{k+1} - f_k} = \frac{1}{2^{\frac{1}{b}} - 1}, \tag{7}$$

The quality factor Q for all filters is constant.

Through this distribution of filters, the transform has a finer frequency resolution in lower frequency range. By an appropriate choice for f_0 and b the center frequencies of the filters correspond directly to musical notes. Therefore this transformation is well suited for the processing of instrumental music, hence the different notes are spaced like the filters in the CQT with $b = 12$ bins per octave or a factor $Q = 16.82$. [6]

In order to achieve a frequency-dependent resolution, the window length therefore has to be chosen in accordance to the analyzed frequency. The desired bandwidth $\Delta_k^{cq} = \frac{f_k}{Q}$ can be realized by choosing a window of length N_k [10].

$$N_k = \frac{f_s}{\Delta_k^{cq}} = Q\frac{f_s}{f_k}, \tag{8}$$

As the factor Q and the sampling frequency f_s are constant, the length of the window varies with f_k. The correlation between the frequency f_k and the window length N_k for different values of Q is shown in Fig. 1.

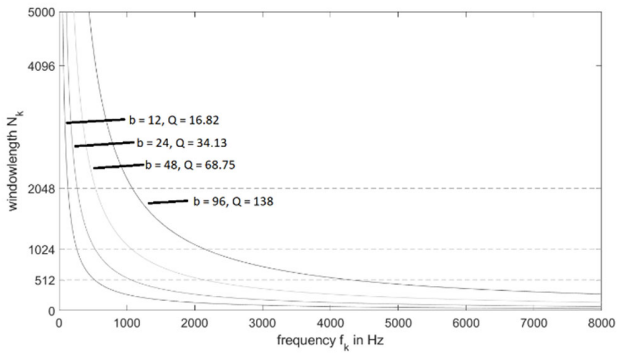

Fig. 1: CQT Window length N_k as a function of f_k for different quality factors

The factor Q is also similar to the number of cycle durations $T_k = \frac{1}{f_k}$ in each window. The distance Δ_k^{cq} between consecutive frequencies f_k and f_{k+1} is crucial for the analysis of a signal. Using the CQT as transform Δ_k^{cq} increases with the frequency while Q remains constant.

Fig. 2 shows the values of Δ_k^{cq} for different CQTs indicated with different parameters b. For comparison, the resolution of the FFT is entered for different lengths N (1024, 2048 and 2048). The used sampling frequency is 44,1 kHz and the base frequency f_0 for the CQT is 20 Hz. The CQT with $b = 96$ maintains a better resolution than the 4096-FFT up to 6 kHz.

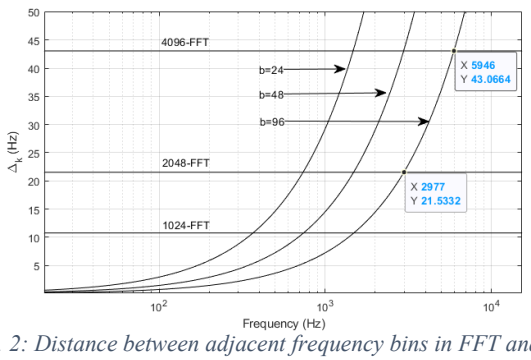

Fig. 2: Distance between adjacent frequency bins in FFT and CQT, $f_s = 44{,}1\ kHz, f_0 = 20\ Hz$

The constant Q transform $X^{cq}(k)$ of a discrete time-domain signal $x(n)$ can be derived from the DFT definition in eq. (3) as follows

$$X^{cq}(k) = \sum_{n=0}^{N_k-1} w_k(n) \cdot x(n) \cdot e^{-j \cdot 2 \cdot \pi \cdot n \frac{f_k}{f_s}} \tag{9}$$

Where $k = 0, \cdots, K-1$ indexes the frequency bins of the CQT and $w_k(n)$ denotes a normalized window function with the length N_k.

Considering equation (8), the constant Q transform $X^{cq}(k)$ can also be evaluated by [6]

$$X^{cq}(k) = \sum_{n=0}^{N_k-1} w_k(n) \cdot x(n) \cdot e^{-j \cdot 2 \cdot \pi \cdot n \frac{Q}{N_k}} \qquad (10)$$

In contrast to the FFT, the window length N_k depends on the frequency and is not constant. The relationship is described in equation (8). For the base frequency f_0, the largest number of samples is required.

Example:

We consider an audio signal sampled at $f_s = 44{,}1\ kHz$. For the CQT we choose $b = 12$ and $f_0 = 20\ Hz$. The quality factor Q as defined in eq. (7) corresponds to approximately 16,82. The window length N_k required for the base frequency f_0 is $N_k \approx 37082$ samples corresponding to $841\ ms$ of the audio signal. This signal duration required for the analysis of the lower frequency range is not suitable for real time applications. It should be noted here that $b = 12$ represents the smallest value for the CQT. If the next level $b = 48$ is set, a signal duration of $3437{,}5\ ms$ is required.

Concerning the computational complexity, the CQT is expensive as compared to the FFT. This is one of the reasons of the unpopularity of this transform, which is partially mitigated by the increased computational computing capacity of modern computers [19]. The asymptotic complexity for the FFT is $\mathcal{O}(N \cdot log_2 N)$, where N is the window length of the input signal. The asymptotic complexity of the CQT following eq. (10) is $\mathcal{O}(N \cdot log_2 N + N \cdot K + K)$, where K represents the number of filters [7].

Because of the large frame length needed for the resolution in lower frequencies, the CQT as defined above is not really suitable for real time applications.

The method proposed in this paper, the Adaptive Quality Frequency Transform (AQFT), limits the window length used for the transform and keeps the logarithmic frequency resolution. The quality factor is adapted to the new length so that the resulting new resolution hardly differs from the CQT frequency division. The time frequency representation is still based on the CQT calculation. Thus, the transform is on the one hand suited for real time applications, because it uses a limited number of samples and on the other hand resembles the frequency resolution of the human ear.

The AQFT algorithm should have several goals:

- reduction of the computational complexity
- controllability of the time resolution for different frequencies
- to keep the deviation from the CQT frequencies low
- controllability of the quality
- compliance with given block lengths

To achieve this some parameter settings are used. The different steps of parameter setting of the proposed transform are illustrated in Fig. 3.

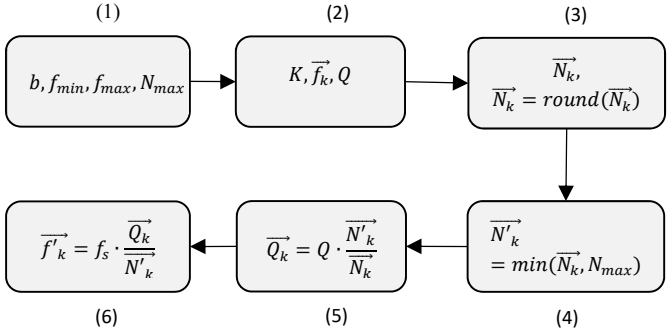

Fig. 3: Steps to determine the parameters of the AQFT

(1) In the beginning, similar to the CQT, the lowest frequency $f_{min} = f_0$, the highest frequency f_{max} and the number of frequency bins per octave b are specified. Further a maximum window length N_{max} is defined.

(2) The total number of frequency bins K (eq. (6)), the discrete frequencies $\overrightarrow{f_k}$ (eq. (5)) and the quality factor Q (eq. (7)) are calculated.

(3) The different window lengths $\overrightarrow{N_k}$ are computed with equation (8). Since the window length describes the number of samples for the analysis, the values for $\overrightarrow{N_k}$ must be integer. Therefore the values are rounded off. Rounding errors arise that influence the quality factor Q and the frequencies $\overrightarrow{f_k}$. The frequency deviation will be corrected later.

(4) The window lengths given in $\overrightarrow{N_k}$ are compared with N_{max}. The smallest value is stored in $\overrightarrow{N'_k}$. $\overrightarrow{N'_k}$ is used for the transform. This step causes a reduction in quality for some frequencies. The term Constant-Q-Transformation can therefore no longer be used.
Fig. 4 shows the window lengths used for the CQT and for the AQFT a function of the frequency f_k.

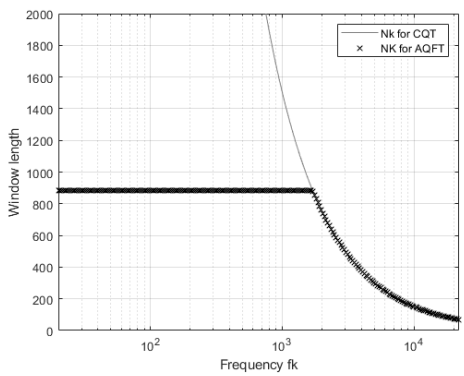

Fig. 4: Window lengths of CQT and AQFT, $b = 24$, $f_0 = 20\ Hz, f_s = 44{,}1\ kHz, N_{max} = 882$ samples

(5) In the next step, the reduced quality is determined so that the frequencies can be retained. Q is adapted to the new length for each frequency. This results in the quality vector $\overrightarrow{Q_k}$. For each bin k and each window length N_k, the quality factor Q_k is computed as follows:

$$Q_k(k) = Q \cdot \frac{N_k'(k)}{N_k(k)} \tag{11}$$

Eq. (11) guarantees that $\overrightarrow{f_k}$ remains unchanged except for small rounding errors.

(6) The small deviations of the frequencies are determined and the frequencies f_k have to be adjusted, so that the frequencies in the spectrum can be interpreted correctly. The new frequencies $\overrightarrow{f_k'}$ are calculated for new window sizes $\overrightarrow{N_k'}$ and new factors $\overrightarrow{Q_k}$.

$$\overrightarrow{f_k'} = \frac{f_s \cdot \overrightarrow{Q_k}}{\overrightarrow{N_k'}} \tag{12}$$

Fig. 5 shows the difference between the CQT and the AQFT frequencies for $f_s = 44{,}1\ kHz, b = 24, f_0 = 20\ Hz$, $f_{max} = \frac{f_s}{2}$, and $N_{max} = 882$ samples ($\cong 20ms$).

The frequencies of the AQFT differ slightly from those of the CQT (gray line). The characteristic division of the frequencies is retained. In this example N_{max} corresponds to 20 ms of the sampled signal. To ensure comparability during the later examination, we have set a block length of a maximum of 20 ms for all transforms.

The difference between the frequency resolution of the AQFT and the CQT resolution Δ_k remains small.

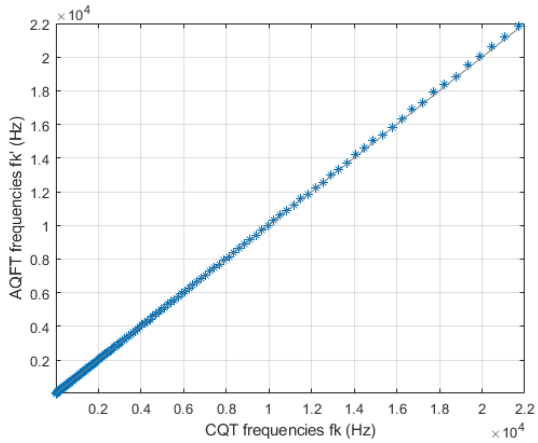

Fig. 5: AQFT vs CQT frequencies; $f_s = 44{,}1\ kHz, b = 24, f_0 = 20\ Hz$; $f_{max} = \frac{f_s}{2}, N_{max} = 882\ samples\ (20ms)$

Equation (10) can still be used to transform the signal using the adapted parameters for each frequency. Since the quality factor is no longer constant but is adapted to the length chosen for each frequency, the resulting transform is called Adaptive Quality Frequency Transform (AQFT).

For effective implementation, we realized the AQFT with a matrix multiplication. Based on eq. (10), a matrix $AQFT(k,n)$ is defined as follows:

$$AQFT(k,n) = w(k,n) e^{-j \cdot 2 \cdot \pi \cdot n \frac{Q}{N(k)}} \tag{13}$$

where $k = 0, \cdots, K-1$ indexes the frequency bins of the CQT and $n = 0, \cdots, N(k) - 1$ the time index. The corresponding Matrix is initialized before the analysis is carried out. That makes the AQFT much more time efficient, since only a matrix multiplication for the analysis has to be carried out (see eq. (14))

$$X(k) = AQFT(k,n) \cdot x(n) \tag{14}$$

Since the window length becomes smaller as the value k increases, some values in these rows of the matrix are zero.

2 THE SIGNAL DERIVATE FFT

The signal derivate FFT (SD-FFT) is a method for high precision fourier analysis of audio signals using signal derivate (SD). This method, proposed in [3] and [4] improves the precision of the power analysis not only in frequency and amplitude but also in time, thus minimizing the problem of the tradeoff of time versus frequency as known with the classical FFT [20].

An improvement of the SD-FFT algorithm used for packet loss recovery in audio signals, is presented in [20]. In addition to the modification of peak picking, a combined FFT/SD-FFT algorithm, depending on the location of the found peaks is introduced. In the present contribution, the SD-FFT is conducted based on the algorithm in [20]. In the following the different steps of the frequency analysis with SD-FFT:

(1) The FFT of the audio signal $x(n)$ and its derivate $x'(n)$ are computed (after windowing with hanning).
(2) Peaks in the magnitude spectrum of $x(n)$ are detected. These peaks are considered to be relevant frequencies in the spectrum.
(3) The amplitudes and frequencies of the SD spectrum are corrected as described in [4] and [3].
(4) The peak frequencies f_i found in $X(k)$ are compared with the SD frequencies $f_{i,SD}$. If the difference between the discrete frequencies (positions in the discrete FFT vector) exceeds one bin (corresponds to the frequency resolution Δf), the algorithm rejects the SD frequencies and takes the FFT frequencies $f_{i,p}$. after adjustment with parabolic regression. Because we are looking for local maxima in the spectrum, the position of a discrete frequency should not be more than one bin away from the peak. Otherwise, that would mean choosing a frequency with a smaller amplitude than the maximum. The exact frequency must in any case be close to the maximum.

3 THE WAVELET PACKET TRANSFORM

The general concept of the wavelet transformation is to divide the input signal into its highpass and lowpass components by scanning it with a so called mother wavelet function. An in depth introduction to wavelets can be found in [21]. This paper will use the Wavelet Packet Transform (WPT), which is a Discrete Wavelet Transform (DWT) with a linear instead of a logarithmic frequency

division. This can be achieved by decomposing the highpass (D, detail) channel of the signal in addition to the procedure for the lowpass (A, approximation) channel as it is done in a DWT. This results in the structure of a complete binary-tree throughout the levels of decomposition [21]. Fig. 6 shows such a tree with two levels. However, in most use-cases it is not necessary to construct the WPT as a complete binary-tree as it leads to more complexity. Therefore, decomposition of the highpass channel is often not fully decomposed to the level of the lowpass channel to achieve a good compromise between frequency resolution and complexity.

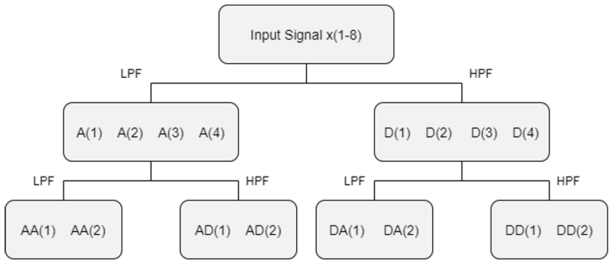

Fig. 6: Wavelet Package Transform with two decomposition levels

An example of an incomplete decomposition tree for usage in cochlea implants can be derived from Nogueira et al. [22].

To decompose the signal, a mother wavelet has to be chosen. Throughout the years, many different mother wavelets have been developed and MATLAB supports various wavelet families [23]. However, as WPT is a discrete transform complex wavelet functions cannot be used because they are only suitable for Continuous Wavelet Transforms (CWT). Nevertheless, this leaves quite a few options such as Daubechies, Symlet, Morlet and many more. The concept to find the best mother wavelet for a given case is by having the best similarity with the signal. Similarity can be defined under different aspects such as energy, entropy or redundancy. Redundancy is, at least for discrete wavelets, not a factor as they do not have any to be measured. Investigations on different wavelets for audio processing for music and speech (where speech could be regarded as a subset of music [17]) have often come to the conclusion that decisions about the chosen wavelet is more about exclusion of worse than picking one best wavelet function as results are quite similar for some "best" [17] and is very dependent on the actual use case [24].

In each level of decomposition, the signal is being convoluted with the corresponding filter coefficients of the mother wavelet. The length of this convolution n is defined as the sum of the length of the input vector v and the order of the filter o. As the following level is expecting an input vector with length of $\frac{v}{2}$, n has to be reduced to the length of v before downsampling. The usage of a circular convolution can help out to avoid the need of such a reduction, as the result of the circular convolution of v and o always has the length of v. Before continuing with the next decomposition level the signal has to be downsampled.

The down sampling on each level of decomposition leads to an aliasing. Therefore, the frequency content of the combination of highpass and lowpass filter (AD) describes a higher frequency content than the combination of two highpass filters (DD). This forces a reordering process of the resulting wavelet coefficients by swapping AD and DD on each level [25]. In MATLAB this can be achieved by using otnodes [26]. The result is shown in Fig. 7.

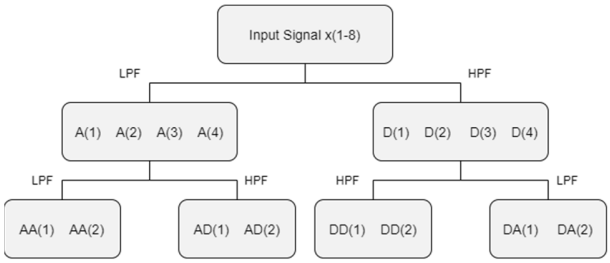

Fig. 7: Frequency ordering in the WPT

After all decompositions have been made, Teager's energy operator is used instead of ordinary energy of the leaf subband to interpret the result. This method for the use in DWT has been proposed by Guido in 2017 [27]. In each leaf subband b the first and last element have to be squared [27]:

$$\tilde{y}_{0,b} \leftarrow (y_{0,b})^2 \tag{15}$$

For leaf subbands with at least two values the following operation has to be executed for any element from the second to the penultimate [27]:

$$\tilde{y}_{i,b} \leftarrow \left(y_{i,b}\right)^2 - \left(y_{i-1,b}\right) \cdot \left(y_{i+1,b}\right) \tag{16}$$

As circular convolution is used to calculate the coefficients of each level the number of elements in each leaf subband b depends on the length of input data l_{Input} and the level of decomposition $level_{dec}$

$$NumOfElements = \frac{l_{Input}}{2^{level_{dec}}} \tag{17}$$

For example, with an input of size 1024 and 8 levels of decomposition the number of elements in each leaf subband is 4.

The resulting coefficients can be used to determine the frequencies in the input signal. The accuracy (resolution in the last level) can be calculated by

$$\Delta f = \frac{f_s}{2 \cdot 2^{level_{dec}}} \tag{18}$$

for the terminal nodes. For $f_s = 16\,kHz$ and a decomposition level of 8 this leads to $\Delta f = 31{,}25\,Hz$.

One of the problems concerning frequency detection with wavelet transformation seems to be the steepness of the wavelet filters. Fig. 8. shows the energy level for each filter of the first two levels of decomposition for a 1.625 kHz sine signal. Channel 1 and 2 show level 1, channel 3 to 6 show level 2. Whilst the segmentation works quite well in level 1, it can be observed that in level 2 of decomposition

a considerable amount of energy is detected in the frequency band of 2-4 kHz which is wrong in terms of frequency detection. As the energy is split up this leads to additional possible frequencies. Furthermore, this leads to a lower amplitude than expected from the original signal.

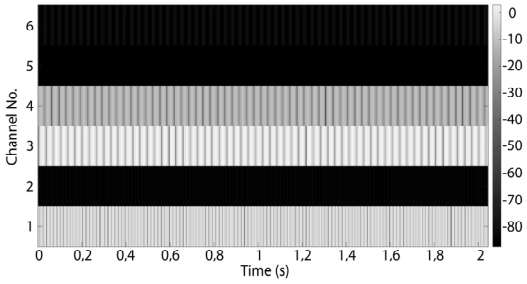

Fig. 8: First two decomposition levels of 'db16'

However, the image above has been taken with a high order mother wavelet 'db16' (Daubechies wavelet of order 16). For lower orders e.g. the often used 'db4' the frequency response is even more flat. Fig. 9 shows the frequency segmentation of the first two decomposition levels with 'db4' of the same sine signal.

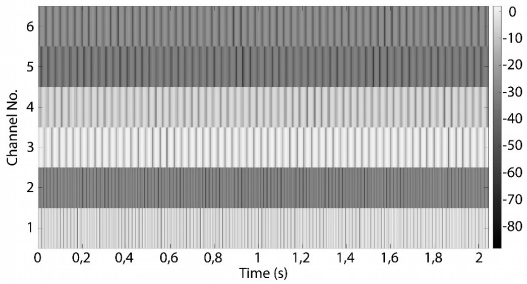

Fig. 9: First two decomposition levels of 'db4'

In this figure the energy is spread to all the different channels of decomposition because of the lack of steepness within the wavelet filter. As common mother wavelets such as Daubechies, Symlet or Meyer (even at higher orders) did not lead to a satisfying result the solution to this problem could be the design of a filter or mother wavelet that has enough steepness to determine the signal frequencies. A guideline to such a filter design can be found in [28].

Another problem is the low frequency resolution of $\Delta f = 31{,}25$ Hz. It can be resolved by using a larger frame with a length of e.g. 8192 results in a resolution of $\Delta f = 1{,}95\ Hz$. However, the total length of signal has to be providing such an extension of the frame by having enough samples. For real-time applications, this length cannot be provided.

Finally, this current implementation of wavelet transformation with 'db16' and 9 decomposition levels does detect the right frequencies within the signal but with wrong amplitudes and additional frequency components.

4 EXPERIMENTAL SETUP AND RESULTS

In this section we will evaluate the performance of the proposed method presented in section I and compare the results with the FFT, SDFFT and PWT. To rate the analysis the frequency-weighted segmental SNR [30] (denoted here by fwSNR) is used, as it considers the frequency and the amplitude of the signal while mimicking the deafness to the phase of the human ear [29]. The processing time required for the transforms mentioned is also compared.

For simulation purpose synthetic files were used. They were created at sampling rates of $16\ kHz$ and $44{,}1\ kHz$ and a duration of 300 ms. The frequency components are located in the first three formants of the human voice [29]. In the first part of the simulation all frequency components are static without any change over the duration of the file. In the second part every frequency component lasts between 10 and 50 ms and shifts in amplitude as well as in frequency value to further emulate the human voice. Fig. 10 and Fig. 11 show respectively spectrograms of a static signal and a dynamic signal.

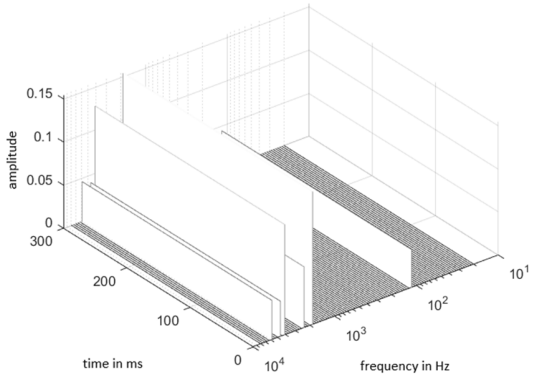

Fig. 10: Spectrogram of a static signal

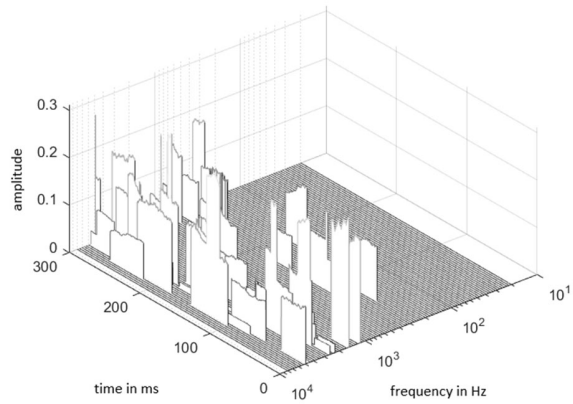

Fig. 11: Spectrogram of a dynamic signal

The investigated transforms are implemented with blocks of 20 ms, which corresponds to 320 samples of $16\ kHz$ signals and 882 samples of $44{,}1\ kHz$ signals. This block length complies in the area of audio signal processing the limits of quasi-stationarity and real-time usability. The result of every analysis is further used to create a spectrogram as shown in the previous figures. Fig. 12 shows an example of the spectrogram created from the results of the FFT analysis, of the sound file from Fig. 11.

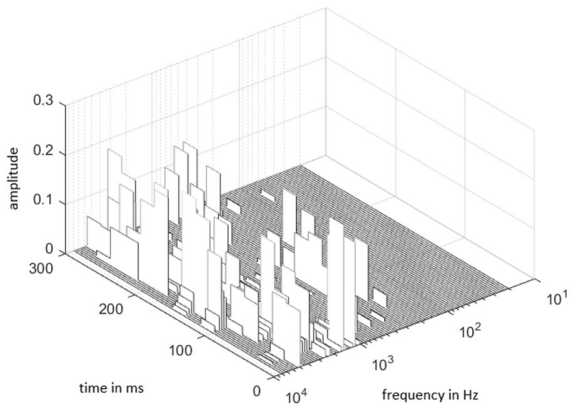

Fig. 12: Spectrogram of a dynamic signal reconstructed from the results of the FFT

Based on the spectrograms generated from the results of the analysis, new audio files are synthesized to evaluate the similarity to the original sound file. To rate the analysis fwSNR is used.

The computations were carried out in MATLAB Simulink on an Intel® Core™ i7-6700HQ CPU at @2.60GHz. The results for 500 static files sampled at 16 kHz are shown in table I and Fig. 13. Table II and Fig. 14 show the results for 500 16 kHz dynamic files.

Table I Simulation results for 500 static files. $N = 320$ samples, $f_s = 16\ kHz$

Transform	Processing time per frame in	fwSNR in dB
FFT	5.313	17.29
SDFFT	3.646	28.96
AQFT (b=24)	2.734	21.97
AQFT (b=48)	3.006	27.12
AQFT (b=96)	3.383	31.79
Wavelet	202.146	-8.911

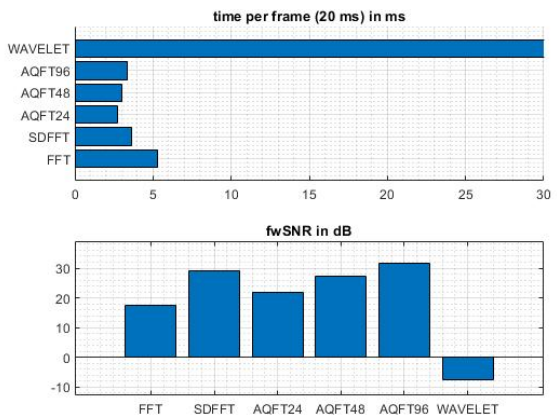

Fig. 13: Results for 500 static files; $f_s = 16\ kHz$

Table II Simulation results for 500 dynamic files. $N = 320$ samples, $f_s = 16\ kHz$

Transform	Processing time per frame in ms	fwSNR in dB
FFT	8.594	8.672
SDFFT	4.818	10.01
AQFT (b=24)	4.75	9.497
AQFT (b=48)	5.078	9.797
AQFT (b=96)	5.339	10.22
Wavelet	219.5	6.892

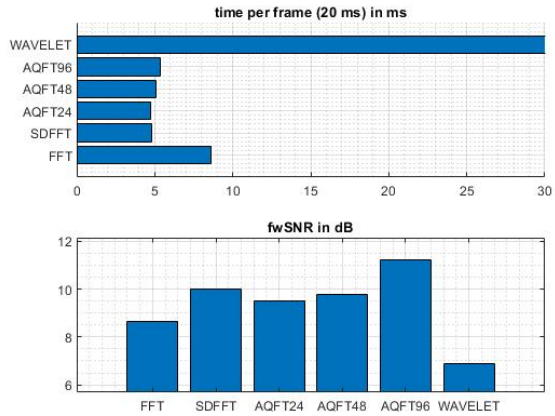

Fig. 14: Results for 500 dynamic files, $f_s = 16\ kHz$

Table III and Fig. 15 show the simulation results for 44,1 kHz static signals. The results for 44,1 kHz dynamic signals are illustrated in table IV and Fig. 16.

Table III Simulation results for 44,1 kHz static files. $N = 882$ samples

Transform	Processing time per frame in ms	fwSNR in dB
FFT	6,66	8,53
SDFFT	3,85	12,93
AQFT (b=24)	5,62	14,63
AQFT (b=48)	6,14	15,73
AQFT (b=96)	8,12	16,45
Wavelet	365,3	2,35

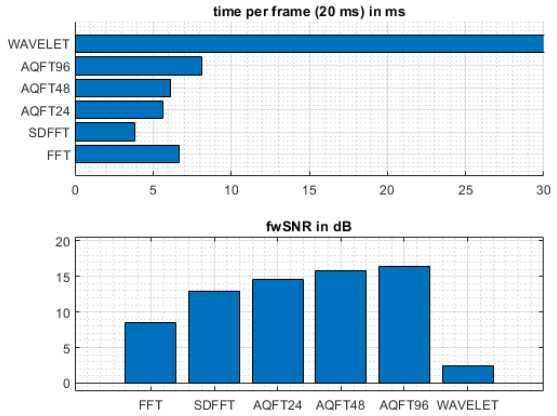

Fig. 15: Results for 44,1 kHz static files; N = 882 samples

Table IV Simulation results for 44,1 kHz dynamic files. $N = 882$ samples

Transform	Processing time per frame in ms	fwSNR in dB
FFT	6,87	5,79
SDFFT	4,48	7,45
AQFT (b=24)	5	9,49
AQFT (b=48)	6,25	10,41
AQFT (b=96)	6,6	10,5
Wavelet	374,16	4,8

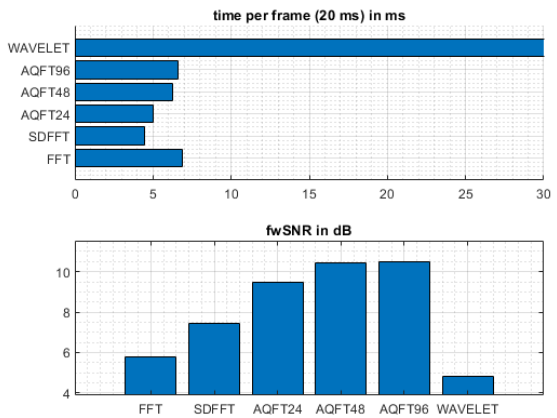

Fig. 16: Results for 44,1 kHz dynamic files; N = 882 samples

The results of these investigations are discussed in section 5.

In addition, we examined the error rate in the detection of frequencies in the implemented procedures. A time-frequency transform can detect frequencies that were not present in the original signal. We implemented this investigation exclusively for static signals. Fig. 17 shows representatively the results for 500 static signals. The values shown describe the mean value over 500 files. For example, with the FFT, a maximum of 13 frequencies in the range from 2413.6 to 9394 Hz were incorrectly detected. These frequencies are not part of the original signal. Incorrectly detected frequencies were only registered with FFT and Wavelet. The error rate increases for higher frequencies. FFT and WPT have a constant frequency resolution. The number of frequencies in the high frequency range is for the human ear unnecessarily high. The probability of wrong detection is therefore higher. With SDFFT, which also has a linear frequency axis, no false frequencies are detected. Only a small frequency and amplitude deviation between original and detected is possible. This is due to the optimizations in the peak-picking algorithm. Due to the logarithmic division of the frequency axis with the AQFT, the probability of false detection remains zero.

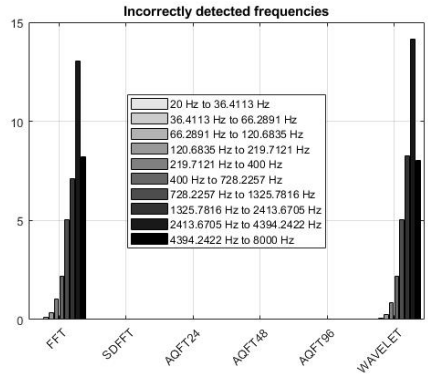

Fig. 17: Incorrectly detected frequencies averaged over 500 static signals.

In the next investigation, we generated a synthetic signal with a frequency division according to the Bark scale. The aim of this investigation is to see how the AQFT behave in the low frequency range compared to FFT, SDFT and PWT. The synthetic signal has a small resolution in the lower frequency range. An overlap of the main lobes in the spectrum is likely. This makes frequency detection based on peak detection more difficult and inaccurate.

Fig. 18 shows an extract for detected frequencies for a synthetic signal with Bark frequency division. The representation is limited to the lower frequency range in order to illustrate the effect of the overlapping of the main peaks.

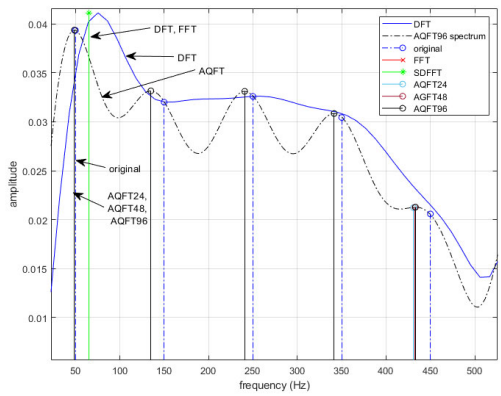

Fig. 18: Detected frequencies for a synthetic signal with Bark frequency division.

The advantage of the AQFT over other transformations is clear. Almost all frequencies are detected here. In contrast to the DFT spectrum, peaks close to the original frequencies can be seen in the AQFT spectrum

The quality assessed with fwSNR shows a clear advantage of the AQFT, especially the AQFT96, compared to the other transforms (see Fig. 19). Concerning the processing time, AQFT gets the best results.

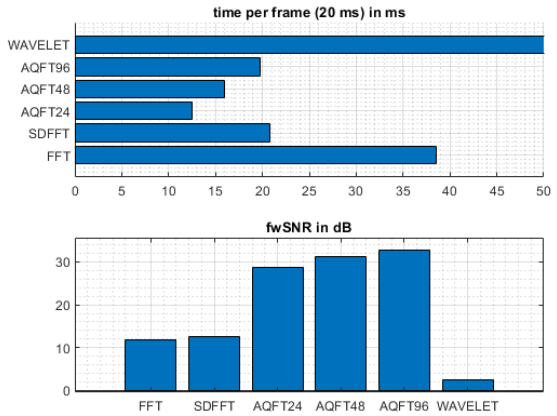

Fig. 19: Results for synthetic file with Bark frequency division. $f_s = 44{,}1\ kHz; N = 882\ samples$

5 DISCUSSION AND CONCLUSIONS

The evaluation of the results achieved in section 4 and a comparison of the processing time and quality of the AQFT and the examined transforms show a significant advantage of the AQFT.

The FFT does not provide the best quality for static as well as dynamic signals and is also not the fastest transformation. It should be noted here that the standard methods of peak picking and parabolic regression are used for the FFT calculation. The implementation of the SDFFT method according to [20] is based on much better routines for determining the valid frequencies. With both static and dynamic signals, the SDFFT proves to be faster and better in quality than the classic FFT. The SDFFT algorithm performs two FFTs per Block, for the signal and for the first derivate. Nevertheless, the processing time of 3.6 ms per 20 ms block for static signals and 4.8 ms for dynamic signals ($f_s = 16\ kHz$) is acceptable. $44{,}1\ kHz$ signals show similar results.

The required processing time by the PWT is far too long compared to other transforms as we are performing a full binary tree wavelet transform. PWT delivers the worst results in comparison with the other transforms in terms of processing time and fwSNR values. This is due to the problems discussed in chapter 3.

Basically, it can be stated that the results of the AQFT and the comparison to other transforms show a similar trend for static and dynamic signals, and for $16\ kHz$ and $44{,}1\ kHz$ signals. The measured fwSNR values are worse with dynamic than with static signals. This is due to the synthesis of the signals on the basis of the spectrograms, which contain frequency jumps.

With an increase in resolution, the processing time of the AQFT rises (Fig. 13 - Fig. 16), as more discrete frequencies are to be considered. The processing time of the AQFT increases proportionally to the selected start quality factor and thus to the complexity of the algorithm. A higher resolution, as realized by AQF96 (b = 96) delivers the best results (fwSNR) for static and dynamic signals and provides a better basis for signal recreation. As a reminder, b describes the number of filters or bins per octave. The processing time required by $16\ kHz$ signals is between 3 ms (static) and 5.5 ms (dynamic) per 20 ms block and thus about 3 ms faster than the FFT. The processing time of the AQFT remains comparable or slightly worse than the SDFFT. This also applies to $44.1\ kHz$ signals.

The Bark signal evaluated in Fig. 19 contains 24 frequencies in each segment, relatively more than the rest of the signals. This leads to a relatively high processing time of approx. 20 ms per 20 ms block.

In the point of view of the accuracy, the AQFT96 reaches the highest fwSNR values for static and dynamic files both for $16\ kHz$ and $44{,}1\ kHz$ signals. Concerning processing time, we see potential for optimization. For the analysis of audio and speech, a hybrid transformation with different b-factors can produce qualitatively similar results with a smaller processing effort. AQFT24 can be sufficient in the high frequency range, with b=96 being selected in the low frequency range up to $4\ kHz$.

An examination of the incorrectly determined frequencies has also shown that probability of detecting wrong frequencies is 0% for both AQFT and SDFFT. Only the amplitude and frequency have a small deviation from the original signal. However, with SDFT it can happen that frequencies cannot be recognized because the peaks in the spectrum are missing due to overlapping. This effect was not observed with AQFT (Fig. 18)

For the transforms to be used in real-time application, the computing hardware and the type of signal to be transformed must be taken into account. If the main purpose is to transform a human speech signal, the AQFT is considered to be the best option as it produces the highest fwSNR value. The computation time as well as the quality are adjustable via the factor b.

6 REFERENCES

[1] D. B. Pisoni und R. E. Remez, Hg., *The handbook of speech perception,* 1. Aufl. Malden, Mass.: Blackwell Publ, 2008.

[2] K. Werner und F. Germain, „Sinusoidal Parameter Estimation Using Quadratic Interpolation around Power-Scaled Magnitude Spectrum Peaks“, *Applied Sciences*, Jg. 6, Nr. 10, S. 306, 2016, doi: 10.3390/app6100306.

[3] M. Desainte-Catherine und S. Marchand, „High-Precision Fourier Analysis of Sounds Using Signal Derivatives*“, *J. Audio Eng. Soc.*, Jg. 48, 7/8, S. 654–667, 2000.

[4] F. Keiler und S. Marchand, „Survey on Extraction of Sinusoids in Stationary Sounds“ in *5th International Conference on Digital Audio Effects DAFx-98*, Hamburg, Germany, 2002, S. 51–58.

[5] J. J. Burred und T. Sikora, „Comparison of frequency-warped representations for source seperation of stereo mixtures“ in *121st Convention Audio, 2006 October 5-8 San Francisco*.

[6] J. C. Brown, „Calculation of a constant Q spectral transform“, *The Journal of the Acoustical Society of America*, Jg. 89, Nr. 1, S. 425–434, 1991, doi: 10.1121/1.400476.

[7] S. Nisar, O. U. Khan und M. Tariq, „An Efficient Adaptive Window Size Selection Method for Improving Spectrogram Visualization“ (eng), *Computational intelligence and neuroscience*, Jg. 2016, S. 6172453, 2016, doi: 10.1155/2016/6172453.

[8] F. Holzmüller, P. Bereuter, P. Merz, D. Rudrich und A. Sontacchi, Hg., *Computational efficient real-time capable constant-Q spectrum analyzer*, 2020. [Online]. Verfügbar unter: http://www.aes.org/e-lib/browse.cfm?elib=20805

[9] C. Schörkhuber, A. Klapuri, N. Holighaus und M. Dörfler, „A Matlab Toolbox for Efficient Perfect Reconstruction Time-Frequency Transforms with Log-Frequency Resolution“ in *AES 53rd International Conference*, London, UK, 2014.

[10] A. Nagathil und R. Martin, „Optimal Signal Reconstruction from a Constant-Q Spectrum“ in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2012*, Kyoto, Japan, 2012, S. 349–352.

[11] C. Schörkhuber und A. Klapuri, „Constant-Q Transform Toolbox For Music Processing“, 2010.

[12] N. Holighaus, M. Dorfler, G. A. Velasco und T. Grill, „A Framework for Invertible, Real-Time Constant-Q Transforms“, *IEEE Trans. Audio Speech Lang. Process.*, Jg. 21, Nr. 4, S. 775–785, 2013, doi: 10.1109/TASL.2012.2234114.

[13] N. Mehala und R. Dahiya, „A Comparative Study of FFT, STFT and Wavelet Techniques for Induction Machine Fault Diagnostic Analysis“ in *7th WSEAS international conference on Computational intelligence, man-machine systems and cybernetics*, Cairo, Egypt, 2008, S. 203–208.

[14] A. Brandolini, A. Gandelli und F. Veroni, „Energy meter testing based on Walsh transform algorithms“ in *IEEE Instrumentation and Measurement Technology Conference - IMTC '94*, Hamamatsu, Japan, 10-12 May 1994, S. 1317–1320, doi: 10.1109/IMTC.1994.351814.

[15] D. V. Anderson, „Speech analysis and coding using a multi-resolution sinusoidal transform“ in *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, Atlanta, GA, USA, 7-10 May 1996, S. 1037–1040, doi: 10.1109/ICASSP.1996.543301.

[16] I.-D. v. d. Nicolae und P.-M. T. Nicolae, „Using the Wavelet Packet Transform to Evaluate Harmonics through a Lookup Table Technique“ in *2016 International Symposium on Fundamentals of Electrical Engineering (ISFEE)*, University Politehnica of Bucharest, Romania, 2016.

[17] F. Bömers, „Wavelets in real time digital audio processing: Analysis and sample implementations“. Masterarbeit, Computer Science IV, University Mannheim, Mannheim, 2000. [Online]. Verfügbar unter: http://www.bomers.de/personal/thesis

[18] J. C. Brown und M. S. Puckette, „An efficient algorithm for the calculation of a constant Q transform“, *The Journal of the Acoustical Society of America*, Jg. 92, Nr. 5, S. 2698–2701, 1992, doi: 10.1121/1.404385.

[19] C. Mateo und J. A. Talavera, „Bridging the gap between the short-time Fourier transform (STFT), wavelets, the constant-Q transform and multi-resolution STFT“, *SIViP*, Jg. 14, Nr. 8, S. 1535–1543, 2020, doi: 10.1007/s11760-020-01701-8.

[20] L. Lajmi, „An Improved Packet Loss Recovery of Audio Signals Based on Frequency Tracking“, *J. Audio Eng. Soc.*, Jg. 66, Nr. 9, S. 680–689, 2018, doi: 10.17743/jaes.2018.0020.

[21] S. D. Panchamkumar, „Complex Wavelet Transforms And Their Applications“. Masterarbeit, Dep. of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, 2003. [Online]. Verfügbar unter: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.102.6504&rep=rep1&type=pdf

[22] W. Nogueira, A. Giese, B. Edler und A. Buchner, „Wavelet Packet Filterbank for Speech Processing Strategies in Cochlear Implants“ in *2006 IEEE International Conference on Acoustics Speed and Signal Processing*, Toulouse, France, 14-19 May 2006, V-121-V-124, doi: 10.1109/ICASSP.2006.1661227.

[23] MathWorks, *Wavelet Families.* [Online]. Verfügbar unter: https://de.mathworks.com/help/wavelet/ug/wavelet-families-additional-discussion.html.

[24] R. J. E. Merry, „Wavelet theory and applications: A literature study“, Dep. of. Mech. Eng., Eindhoven University of Technology, Eindhoven, 2005. [Online]. Verfügbar unter: http://www.mate.tue.nl/mate/pdfs/5500.pdf

[25] X.-w. Zeng, W.-m. Zhao und J.-q. Sheng, „Corresponding relationships between nodes of decomposition tree of wavelet packet and frequency bands of signal subspace“, *Acta Seismol. Sin.*, Jg. 21, Nr. 1, S. 91–97, 2008, doi: 10.1007/s11589-008-0091-x.
[26] MathWorks, *otnodes.* [Online]. Verfügbar unter: https://de.mathworks.com/help/wavelet/ref/otnodes.html.
[27] R. C. Guido, „Effectively Interpreting Discrete Wavelet Transformed Signals [Lecture Notes]“, *IEEE Signal Process. Mag.*, Jg. 34, Nr. 3, S. 89–100, 2017, doi: 10.1109/MSP.2017.2672759.
[28] S. R. M. Penedo, M. L. Netto und J. F. Justo, „Designing digital filter banks using wavelets“, *EURASIP J. Adv. Signal Process.*, Jg. 2019, Nr. 1, 2019, doi: 10.1186/s13634-019-0632-6.
[29] B. Pfister und T. Kaufmann, *Sprachverarbeitung: Grundlagen und Methoden der Sprachsynthese und Spracherkennung*. Berlin, Heidelberg: Springer-Verlag, 2008.
[30] Arkadiy Prodeus, Vitalii Didkovskyi, Maryna Didkovska, Igor Kotvytskyi, *2018 International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S & T): Objective and Subjective Assessment of the Quality and Intelligibility of Noised Speech*. [Place of publication not identified]: IEEE, 2018.

7 NOMENCLATURE

FFT	=	Fast Fourier Transform
STFT	=	Short Time Fourier Transform
SDFFT	=	Signal Derivate FFT
CQT	=	Constant Quality Transform
AQFT	=	Adaptive Quality Fourier Transform
DWT	=	Discrete Wavelet Transform
PWT	=	Packet Wavelet Transform
fwSNR	=	Frequency-weighted Signal to Noise Ratio