# COMPUTATIONAL COST OF CHIRP Z-TRANSFORM AND GENERALIZED GOERTZEL ALGORITHM

*Pavel Rajmic* *Zdenek Prusa*[⋆] *Christoph Wiesmeyr*[†]

Dept. of Telecommunications, FEEC, Brno University of Technology,
Technická 12, 616 00 Brno, Czech Republic, `rajmic@feec.vutbr.cz`
[⋆]Acoustics Research Institute, Austrian Academy of Sciences,
Wohllebengasse 12–14, 1040 Wien, Austria, `zdenek.prusa@oeaw.ac.at`
[†] Numerical Harmonic Analysis Group, Faculty of Mathematics, University of Vienna
Oskar-Morgenstern-Platz 1, 1090 Wien, Austria, `christoph.wiesmeyr@univie.ac.at`

## ABSTRACT

Two natural competitors in the area of narrow-band spectrum analysis, namely the Chirp Z-transform (CZT) and the Generalized Goertzel algorithm (GGA), are taken and compared, with the focus on the computational cost. We present results showing that for real-input data, the GGA is preferable over the CZT in a range of practical situations. This is shown both in theory and in practice.

***Index Terms*—** Generalized Goertzel Algorithm, Chirp Z-transform, spectrum analysis, computational complexity, comparison, speed

## 1. INTRODUCTION

A signal processing engineer is often interested in computing the spectrum of a small band of interest at a fine resolution. This applies both to amplitude (more often) and phase. The standard FFT provides too coarse spectral information and is global.

The Chirp Z-transform algorithm [1] (CZT) is a well-established and widely used method for this type of spectral analysis. The Generalized Goertzel Algorithm [2] (GGA) can be utilized as well, since the generalization allows for arbitrary, non-integer frequency indexes, in contrast to the standard Goertzel algorithm (and the DFT). Since the GGA has been introduced only recently, the respective computational costs have not been compared in the literature yet.

Of course, there are other methods of obtaining the desired spectral samples: First, we can simply zero-pad the signal and run a standard FFT on this new sequence to get a fine-resolved spectrum. However, with this approach the vast majority of computed coefficients are of no interest. Methods such as the pruned-FFT [3, 4] can be applied to reduce the computational cost, but their performance, roughly speaking, heavily depends on the divisibility properties of the number of frequency bins and the signal length. As a second option we mention the Zoom-FFT [5]. As presented in [6], this algorithm requires a number of steps including modulation, low-pass filtering and decimation, which make the algorithm more costly than both the CZT and the GGA.

The goal of this paper is to analyze the complexities of the CZT and the GGA and find conditions under which one is preferable to the other. Throughout the rest of this contribution we assume only *real input* signals.

In Sections 2 and 3 we briefly review the two algorithms. The theoretical comparison is done in Sec. 4.1, while in Sec. 4.2 we present numerical experiments. Some other aspects are then treated in Sec. 4.3.

## 2. GENERALIZED GOERTZEL ALGORITHM

The original algorithm invented by G. Goertzel [7], in the following abbreviated to GA, serves to compute the $k$-th single DFT entry for the signal $\{x[n]\}$ of length $N$, i.e.

$$X[k] = \sum_{n=0}^{N-1} x[n]\,\mathrm{e}^{-\mathrm{j}2\pi k \frac{n}{N}}, \quad k = 0, \ldots, N-1. \qquad (1)$$

This equation can, of course, be used directly for obtaining the DFT values, but the GA is advantageous because it requires a quarter less computations [8]. Multiplying the right side of equation (1) by $1 = \mathrm{e}^{\mathrm{j}2\pi k \frac{N}{N}}$ leads to its equivalent,

$$X[k] = \sum_{n=0}^{N-1} x[n]\,\mathrm{e}^{-\mathrm{j}2\pi k \frac{n-N}{N}}, \qquad (2)$$

which could be regarded as a convolution. Standard signal processing literature then helps to rewrite this as an second-order IIR system. Denoting its state variables by $s$, the desired

This work was supported by projects CZ.1.05/2.1.00/03.0072, CZ.1.07/2.3.00/20.0094 and CZ.1.07/2.2.00/28.0062 co-financed by the European Social Fund and the state budget of the Czech Republic, and partly supported by the Austrian Science Fund (FWF) START-project FLAME Y 551-N13, as well as FWF NFN S106 grant SISE.

spectral sample $X[k]$ is found as the output of such a system at time $N$:

$$X[k] = s[N] - \mathrm{e}^{-\mathrm{j}\frac{2\pi k}{N}} s[N-1]. \tag{3}$$

It should be emphasized that the transition from (1) to (2) holds for integer-valued $k$ only; in the case of $k \in \mathbb{R}$, these two formulas are generally no longer in agreement. In fact, when $k$ is not integer-valued, we should speak of the Discrete-time Fourier transform (DTFT), defined by

$$X(\omega) = \sum_{n=-\infty}^{\infty} x[n]\,\mathrm{e}^{-\mathrm{j}\omega n}, \ \omega \in \mathbb{R}. \tag{4}$$

With the notation $\omega_k = 2\pi\frac{k}{N}$, $k \in \mathbb{R}$, we can write that

$$X(\omega_k) = \sum_{n=0}^{N-1} x[n]\,\mathrm{e}^{-\mathrm{j}2\pi k\frac{n}{N}}, \tag{5}$$

where we exploited the compactness of the support of the signal $\{x[n]\}$.

The generalized Goertzel algorithm (GGA) [2] is applicable for any $k \in \mathbb{R}$: First, (5) is extended by unity in the form of $\mathrm{e}^{\mathrm{j}2\pi k\frac{N}{N}} \cdot \mathrm{e}^{-\mathrm{j}2\pi k\frac{N}{N}} = 1$ for $k \in \mathbb{R}$, leading to

$$X(\omega_k) = \mathrm{e}^{-\mathrm{j}2\pi k} \sum_{n=0}^{N-1} x[n]\,\mathrm{e}^{\mathrm{j}2\pi k\frac{N-n}{N}}. \tag{6}$$

Now with the same techniques as in the standard case we arrive at the DTFT coefficient in the form of

$$X(\omega_k) = \left(s[N] - \mathrm{e}^{-\mathrm{j}\frac{2\pi k}{N}} s[N-1]\right) \cdot \mathrm{e}^{-\mathrm{j}2\pi k}. \tag{7}$$

Comparing this with the above, we indeed see that it is a generalization, since the constant $\mathrm{e}^{-\mathrm{j}2\pi k}$ equals one for $k \in \mathbb{Z}$. In fact, the only variation compared to the standard Goertzel algorithm is the multiplication by this constant at the very end of the algorithm. Clearly, $\mathrm{e}^{-\mathrm{j}2\pi k}$ affects only the phase of $X(\omega_k)$. We note that part of the community are aware of the possibility of using the standard GA also for $k \notin \mathbb{Z}$ [6], but to the best of our knowledge, this is only performed in the context of measuring the *amplitude*, which is clearly not affected, in contrast to the *phase*.

It is shown in [2] that the algorithm (in fact, both the standard and the generalized) can be further shortened by a few computations. The shortened GGA is summarized in Fig. 1.

The GGA has been recently exploited as a fine-scale spectrum analyzer for the detection of damages in cantilever beams [9].

## 3. CHIRP Z-TRANSFORM ALGORITHM

The Chirp-Z transform, described well in a number of sources [1, 8, 10], is a procedure used to compute a limited range of spectral frequencies, which are linearly spread over a particular range. Formally, we are interested in $K$ spectral samples $\omega_k = \omega_0 + k\Delta\omega$, $k = 0, \ldots, K-1$, i.e.

$$X(\omega_k) = \sum_{n=0}^{N-1} x[n]\mathrm{e}^{-\mathrm{j}(\omega_0 + k\Delta\omega)n}. \tag{8}$$

Substituting $W = \mathrm{e}^{-\mathrm{j}\Delta\omega}$ and using a trick due to Bluestein [11] yields

$$X(\omega_k) = W^{\frac{k^2}{2}} \sum_{n=0}^{N-1} x[n]\mathrm{e}^{-\mathrm{j}\omega_0 n} W^{\frac{n^2}{2}} W^{-\frac{(k-n)^2}{2}} \tag{9}$$

for $k = 0, \ldots, K-1$, which can be treated as the convolution of two sequences, namely $\{x[n]\mathrm{e}^{-\mathrm{j}\omega_0 n} W^{\frac{n^2}{2}}\}$ and $\{W^{-\frac{n^2}{2}}\}$, followed by multiplication by $W^{\frac{k^2}{2}}$. The name of the transform comes from the fact that the signal $\{W^{-\frac{n^2}{2}}\}$ is usually called a (linear) *chirp*. Padding these sequences to a proper length (power of two in common implementations) allows the computation of (9) via fast convolution, i.e. via multiplication in the spectral domain using FFT. The steps of the algorithm are detailed in Fig. 2.

```
**Input:** frequency "index" $k \in \mathbb{R}$; signal $x$ of length $N$
**Output:** $y$, representing $X(\omega_k)$ according to eq. (5)

%Precalculation of constants
$A = 2\pi\frac{k}{N}$, $B = 2\cos A$, $C = \mathrm{e}^{-\mathrm{j}A}$, $D = \mathrm{e}^{-\mathrm{j}\frac{2\pi k}{N}(N-1)}$
%State variables
$s_0 = 0$, $s_1 = 0$, $s_2 = 0$
%Main loop
for $i = 0 : N-2$    %one iteration less than traditionally
  $s_0 = x[i] + B \cdot s_1 - s_2$
  $s_2 = s_1$
  $s_1 = s_0$
end
%Finalizing calculations
$s_0 = x[N-1] + B \cdot s_1 - s_2$
$y = s_0 - s_1 \cdot C$
$y = y \cdot D$    %constant substituting the iteration $N-1$, and correcting
the phase at the same time
```

**Fig. 1**. Generalized Goertzel algorithm with shortened iteration loop. The changes, compared to the standard Goertzel algorithm, are marked in color.

## 4. COMPARING GGA AND CZT

We first list a few basic, general facts about both algorithms and then focus on quantifying the computational complexity.

- The GGA allows computations on the run, i.e. each time a single new sample is acquired, immediate updating of the state variables is possible. The CZT, on the other hand, can only start when all the samples have been received.
- While the signal length $N$ is usually pushed to be a power of two for maximum FFT performance, the complexity of the GGA grows linearly in $N$ (for $K$ fixed).

```
**Input:** starting frequency $\omega_0 \in \mathbb{R}$; frequency step $\Delta\omega$; number of frequencies $K$; signal $x$ of length $N$;
$N_{\mathrm{FFT}} = \mathrm{nextfftlength}\,(N + K - 1)$
**Output:** Array $y$ of $K$ complex numbers representing $X(\omega_k)$ according to Eq. (8)

%Precalculation
Form sequence $c[n] = W^{n^2/2}$ for $n = 0, \dots, K - 1$
Form sequence $d[n] = \mathrm{e}^{-\mathrm{j}\omega_0 n} c[n]$ for $n = 0, \dots, N - 1$
Form sequence of length $N_{\mathrm{FFT}}$:
$h[n] = W^{-n^2/2}$ for $n = 0, \dots, K - 1$,
$h[N_{\mathrm{FFT}} + n] = W^{-n^2/2}$ for $n = -(N - 1), \dots, -1$,
and zeros elsewhere.
Pad $x$ with zeros to length $N_{\mathrm{FFT}}$.
%Main computation
$g[n] = x[n]d[n]$ for $n = 0, \dots, N_{\mathrm{FFT}} - 1$
$z[n] = g[n] \circledast h[n]$ via fast convolution, i.e. using three FFTs
%Postcalculation
$y[n] = z[n]c[n]$ for $n = 0, \dots, K - 1$
```

**Fig. 2**. Steps of the basic Chirp Z-transform algorithm. Function $\mathrm{nextfftlength}\,(N + K - 1)$ returns the next fast FFT length.

- The GGA is more flexible in choosing the frequencies, we can for example vary the sampling density in the interval of interest. This cannot be achieved by the CZT.
- Since the GGA is implemented as an IIR filter, large $N$ results in a propagation of the quantization error, thus in the decrease of the accuracy [12].

It should also be noted that the general definition of CZT no longer assumes the complex kernel to lie on the unit circle as in the case in (8). The form of CZT restricted to the unit circle is sometimes (improperly) referred to as the Fractional Fourier Transform (FrFT) [13]. Dropping this assumption, however, makes no difference to our analysis, since such a generalization is possible for both the presented algorithms at a negligible cost increase.

### 4.1. Computational cost in theory

The complexity of the radix-2 FFT is assumed to be $\alpha N \log_2 N$ in all the quantifications, where $\alpha$ ranges around the interval from 4 to 5 for real inputs, depending on the implementation.

It can be easily shown [2] that for real input data of length $N$ the GGA requires $3N$ operations per extracted frequency, which leads to an overall flop count of

$$F_{\text{GGA}} = 3NK. \tag{10}$$

In specific situations, it is possible to further simplify the computations, based on combination of the FFT and Goertzel, see e.g. [4], which is beyond the scope and intent of the paper.

Should the spectral analysis via (8) be evaluated for $K$ frequencies in this direct form, it would cost $8KN$ operations. The flop count of the "fast" Chirp-Z transform is given by $4[\alpha N \log_2 N + (\alpha + 6)N]$, which holds for the case $K = N$. This result can be obtained adopting the analysis from [13]. For $K < N$ it is possible to count the flops to

$$F_{\text{CZT}} = 4\alpha N \log_2 K + (4\alpha + 25)N - K \tag{11}$$

using a decimation/factorization scheme, under the assumption that $K$ divides $N$ and $N$ is a power of 2. If $K$ and $N$ do not fulfill these conditions, the computational cost increases.

In the following we will compare the fast CZT, which computes the $K$ spectral samples at once, to obtaining the samples individually using the GGA $K$ times starting from $\omega_0$ and hopping by $\Delta\omega$ steps. Using the computational complexities stated in formulas (10) and (11) we find the GGA to be more efficient if $3KN < 4\alpha N \log_2 K + (4\alpha + 25)N - K$. Dividing this inequality by $N$ and neglecting the term $K/N$, since we assume $K \ll N$, lead to

$$K < \frac{1}{3}\left(4\alpha \log_2 K + 4\alpha + 25\right). \tag{12}$$

The crossover point is approximately $K = 53$ for $\alpha = 5$ and $K = 42$ for $\alpha = 4$, under the above-mentioned circumstances. It is interesting to observe that this quantity is *independent of the signal length $N$*. Concluding, the theory suggests choosing the CZT if the frequency grid contains more than about 50 points. If one is interested in computing fewer spectral samples, then the GGA is more advantageous.

### 4.2. Computational cost in practice

The GGA implementation is a straightforward transcription of the pseudo code in Fig. 1. The only trick used is a manual unrolling of the outer loop over the required frequencies $k$, i.e. it is divided into small chunks of length $M$ and the algorithm then uses $M$ sets of state variables and constants. There are two versions of GGA implemented: with (GGA curve) and without (GGA(np)) precomputing the complex exponentials.

The CZT was implemented both directly as described in Sec. 3 (CZT curves), and using the factorization/decimation scheme [13] (CZT(fact), not described in this paper) using the FFTW library [16]. In both cases all the chirps and frequency responses possible were precomputed together with the FFTW plans and as such they are not encompassed in the measurements. We did so in order to make the running times comparable. The precomputing procedure in the CZT algorithms implementations usually takes more than ten times the duration of the actual computation even with the `FFTW_ESTIMATE` flag in FFTW plans.

Both algorithms for computing the CZT use padding to a next length suitable for the FFT. Since the FFTW library contains implementation of FFT algorithms not restricted to the powers of two, but based on powers of small prime numbers, we compare the implementations that use padding to the next power of two (subscript nextpow2) and to the next lengths with factors 2, 3 and 5 (subscript nextfastfft) [14].

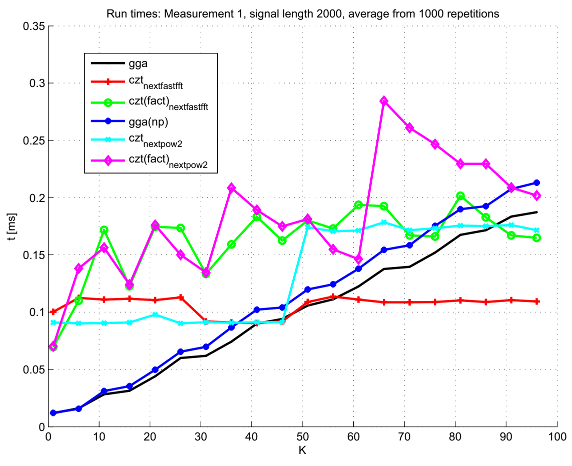

**Fig. 3**. An example of absolute timing result on a single computer. Signal length 2000.

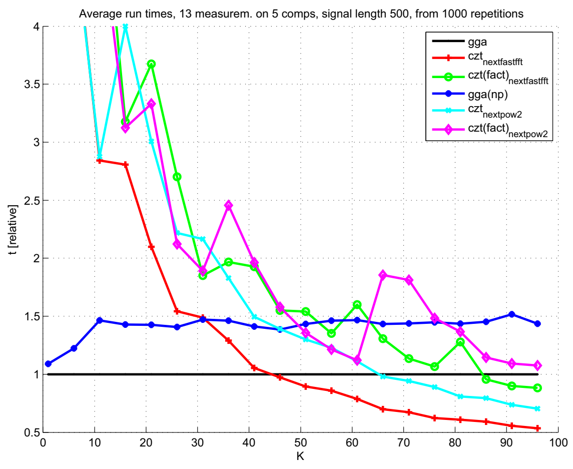

**Fig. 4**. An example of relative timing results (the GGA being the anchor), average over five computers, signal length 500.

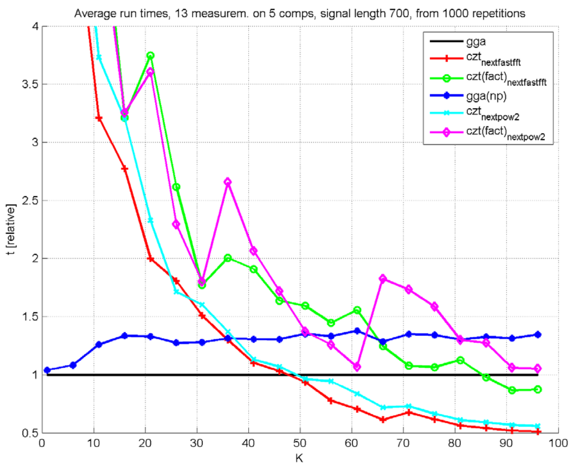

**Fig. 5**. An example of relative timing results (the GGA being the anchor), average over five computers, signal length 700.

*Implementation details*

The benchmark programs were written in the C language (standard C99) and compiled as 64-bit Windows binaries using GCC 4.8.1 from the MinGW system provided by the TDM-GCC compiler suite [15]. The FFT was computed using the FFTW 3.3.3 library; more precisely, we used the precompiled 64-bit binaries version obtainable from the FFTW homepage. All FFTW plans were created using the `FFTW_MEASURE` flag, all inputs were real `double` data type. The compilation was done using the `-O3` optimization parameter.

The source codes of the benchmark programs are available at the paper webpage [17]. Both the C and the Matlab source codes of the algorithms are part of the LTFAT [18, 19] beginning with version 1.4.4.

*Results*

We show a particular measurement in Fig. 3. We can see here that the GGA complexity grows linearly (in both variants), in accordance with the theoretic calculation. The timing for the CZT is greatly dependent on the sum $N+K$. For example, the abrupt step in the $\mathrm{CZT}_{\mathrm{nextpow2}}$ curve is due to moving to the next power of two at about 2048 samples. However, the point where the compared methods cross each other is observable in the nearness of $K = 50$ as suggested by the theory.

In Fig. 4, we present the relative results, averaged, in particular for a signal length of 500 samples. The same type of plot is presented in Fig. 5, but for signal 700 samples long.

Due to the limited space for this contribution, we invite the interested reader to visit the webpage [17] to see more graphs, the computer configurations, etc. The crosspoints for the tested lengths $N \in \{300, 500, 700, 1024, 1300, 1700, 2000\}$ all fall into the range from $K = 40$ to $K = 60$. Generally, CZT including the nextfastfft lenght selection was better than the standard nextpow2 implementation. Interestingly, both the tested CZT(fact) algorithms perform worse than the simpler methods that make no use of decimation/factorization.

We are aware of the fact that this study is performed only on a fraction of possible computing system architectures. On platforms (systems) other than PC (with MS Windows), the results could differ, and this difference would be credited mainly to the speed of the FFT (FFTW), since this is the crucial part of the CZT.

## 4.3. Memory aspects

It should be noted that the GGA in comparison with the CZT algorithms requires very little additional memory. In the precomputed version a complex array of length $K$ is formed, and the direct version requires only storing the state variables! In contrast, the CZT in its direct form requires storing $K + N + N_{\text{FFT}}$ plus memory for storing FFTW plan(s), not counting the buffer for the FFT, but some memory can be traded off for the increased execution time.

## 5. CONCLUSION

The paper shows that there is consistency between the theoretic and the practical comparison of the CZT and the GGA algorithms used for fine-scale spectrum analysis. In our setup, the GGA is advantageous over the CZT for number of analyzed frequencies not exceeding ca $K = 50$.

A natural idea comes consequently: since the GGA can be parallelized trivially, on parallel systems it can beat the CZT easily. But on the other hand, the CZT in its factored form can also be made parallel from part. What will, in the end, be the leading influence in the comparison depends on too many factors to make a simple conclusion.

## REFERENCES

[1] L. R. Rabiner, R. Schaffer, and C. M. Rader, "The chirp z-transform algorithm," *IEEE Transactions on Audio and Electroacoustics*, vol. 17, pp. 86–92, 1969.

[2] Petr Sysel and Pavel Rajmic, "Goertzel algorithm generalized to non-integer multiples of fundamental frequency," *EURASIP Journal on Advances in Signal Processing*, vol. 56, 3 2012.

[3] Ryszard Stasinski, "FFT pruning – a new approach," in *EUSIPCO-86 Signal procesing III: Theories and applications*. 1986, vol. 1, pp. 267–270, North-Holland.

[4] H.V. Sorensen and C.S. Burrus, "Efficient computation of the DFT with only a subset of input or output points," *IEEE Transactions on Signal Processing*, vol. 41, no. 3, pp. 1184 –1200, 1993.

[5] N. Thrane, "Zoom-FFT," Tech. Rep. 2, Bruel&Kjaer, 1980, ISSN 0007-2621.

[6] R. G. Lyons, *Understanding digital signal processing*, Prentice Hall PTR, New Jersey (USA), 2 edition, 2004.

[7] G. Goertzel, "An algorithm for the evaluation of finite trigonometric series," *American Mathematical Monthly*, vol. 65, no. 1, pp. 34–35, 1958.

[8] A V Oppenheim, R W Schafer, and J R Buck, *Discrete-time signal processing*, Prentice-Hall, New Jersey (USA), 2 edition, 1998.

[9] Darian M. Onchis and Pavel Rajmic, "Generalized Goertzel algorithm for computing the natural frequencies of cantilever beams," *Signal Processing*, vol. 96, pp. 45–50, 3 2014.

[10] John G. Proakis and Dimitris G. Manolakis, *Digital signal processing*, Prentice Hall, 1996.

[11] Leo I. Bluestein, "A linear filtering approach to the computation of the discrete Fourier transform," *IEEE Transactions on Audio and Electroacoustics*, vol. 18, pp. 451–455, 1970.

[12] Alicja Smoktunowicz and Iwona Wrobel, "On improving the accuracy of Horner's and Goertzel's algorithms," *Numerical Algorithms*, vol. 38, no. 4, pp. 243–258, 2005.

[13] D. H. Bailey and P. N. Swarztrauber, "The fractional Fourier transform and applications," *SIAM Review*, vol. 33, pp. 389–404, 1991.

[14] Peter Lempel Søndergaard, "LTFAT-note 17: Next fast FFT size," Tech. Rep., Technical University of Denmark, 2011, URL: `http://ltfat.sourceforge.net/notes/ltfatnote017.pdf`.

[15] "TDM-GCC compiler suite for windows webpage," 2013, URL: `http://tdm-gcc.tdragon.net`.

[16] Matteo Frigo and Steven G. Johnson, "The design and implementation of FFTW3," *Proceedings of the IEEE*, vol. 93, no. 2, pp. 216–231, 2005, Special issue on "Program Generation, Optimization, and Platform Adaptation".

[17] Pavel Rajmic, "Supplementary material to Computational cost of Chirp Z-transform and Generalized Goertzel algorithm," 2013, URL: `http://www.utko.feec.vutbr.cz/~rajmic/gga/gga_vs_czt.html`.

[18] Peter L. Søndergaard, Bruno Torrésani, and Peter Balazs, "The Linear Time Frequency Analysis Toolbox," *International Journal of Wavelets, Multiresolution Analysis and Information Processing*, vol. 10, no. 4, 2012.

[19] Peter Lempel Søndergaard, "LTFAT webpage," 2013, URL: `http://ltfat.sourceforge.net`.