

# FINDING LINE SPECTRAL FREQUENCIES USING THE FAST FOURIER TRANSFORM

Tom Bäckström<sup>1,2</sup>, Christian Fischer Pedersen<sup>1,3</sup>, Johannes Fischer<sup>1</sup> and Grzegorz Pietrzyk<sup>2</sup>

<sup>1</sup> International Audio Laboratories Erlangen, Friedrich-Alexander University (FAU)

<sup>2</sup> Fraunhofer IIS, Erlangen, Germany. <mailto:tom.backstrom@audiolabs-erlangen.de>

<sup>3</sup> Electrical and Computer Engineering, Department of Engineering, Aarhus University, Denmark.

## ABSTRACT

Main-stream speech codecs are based on modelling the speech source by a linear predictor. An efficient domain for quantization and coding of this linear predictor is the line spectral frequency representation, where the predictor is encoded into an ordered set of frequencies that correspond to the roots of the corresponding line spectral polynomials. While this representation is robust in terms of quantization, methods available for finding the line spectral frequencies are computationally complex. In this work, we present a method for finding these frequencies using the FFT, including methods for limiting numerical range in fixed-point implementations. Our experiments show that, in comparison to a zero-crossing search in the Chebyshev domain, the proposed method reduces complexity and improves robustness, while retaining accuracy.

**Index Terms**— speech coding, line spectral frequencies, linear prediction, root finding

## 1. INTRODUCTION

The most frequently used paradigm in speech coding is Algebraic Code Excited Linear Prediction (ACELP), which is used in standards such as the AMR-family, G.718 and MPEG USAC [1–3] as well as the recent 3GPP Enhanced Voice Services standard [4]. It is based on modelling speech using a source model, consisting of a linear predictor (LP) to model the spectral envelope, a long time predictor (LTP) to model the fundamental frequency and an algebraic codebook to represent the residual.

The coefficients of the linear predictive model are very sensitive to quantization, whereby they are usually first transformed to Line Spectral Frequencies (LSFs) or Immitance Spectral Frequencies (ISFs) before quantization. The LSF/ISF domains are relatively robust to quantization errors and the stability of the predictor can, in these domains, be readily preserved, whereby it offers a suitable domain for quantization [5–7].

The line spectral frequency representation is based on a linear transformation of the linear predictor into a pair of

polynomials, known as the line spectrum pair polynomials, which have their roots on the unit circle, whereby their locations can be described by their angles or frequencies only. Hence, the roots of the line spectrum pair polynomials are known as the line spectral frequencies of the corresponding predictor [7].

While the representation is in itself robust and gives control of both the stability and, to some extent, perceptual features of the predictor [8], finding the line spectral frequencies is a non-trivial task. It is essentially a polynomial root-finding problem, for which no analytic solution exists when the order of the polynomial is in the conventional range.

An early approach by Kang [9] for finding the line spectral frequencies was based on evaluating the spectral magnitude of the spectrum with the FFT. Since the roots of the two polynomials are on the unit circle, they will appear as valleys in the magnitude spectrum. By searching for valleys of the power or magnitude spectrum, we can thus locate all the roots, provided that the spectrum is sampled with a sufficiently high accuracy such that all valleys are visible. A benefit of this approach is that it is simple and it applies the FFT, which is known to be both fast and numerically stable.

The approach of Kang does not, however, take benefit of the fact that the line spectral polynomials are symmetric and antisymmetric, respectively, and that they have real-valued coefficients, whereby their roots appear in complex-conjugate pairs. An approach by Kabal [10] applies the Chebyshev transform on the line spectral polynomials, to simultaneously remove these redundancies and project complex conjugate roots onto the real axis. Since the transformed polynomials are half the order and their roots are on the real axis, we can estimate their locations with a simple zero-crossing search. This approach is applied for example in AMR and G.718 [1, 2].

The benefit of the approach of Kabal is the reduction in the order of the polynomial, which reduces complexity, but unfortunately the Chebyshev transform concurrently also prohibits the use of the FFT. In addition, since we can readily show that the Chebyshev transform is not an orthonormal transform, for higher order polynomials ( $m > 32$ ), the numerical accuracy rapidly degrades. The approach can therefore not be applied on higher-order polynomials.

Pedersen performed the work while employed by FAU.

In this work, we present a method similar to Yedlapalli's method [11], which applies a zero-crossing search in the Fourier spectrum, whereby we can take advantage of the efficient FFT algorithm, while simultaneously retaining the accuracy of Kabal's approach (see Section 2 for details). While Yedlapalli's method requires high numerical accuracy [11], we apply a pre-processing step in Section 3 which limits numerical range of spectral coefficients. It follows that the computational complexity of a fixed-point implementation is reduced. Our experiments in Section 4 show that the proposed method has a significantly lower computational complexity and improved accuracy in comparison to G.718.

## 2. ROOT SEARCH IN THE SPECTRAL DOMAIN

Given the Z-transform of a linear predictive model  $A(z) = \sum_{k=0}^m a_k z^{-k}$  of order  $m$ , the line spectral polynomials are defined as [7]

$$\begin{cases} P(z) &= A(z) + z^{-m-l}A(z^{-1}) \\ Q(z) &= A(z) - z^{-m-l}A(z^{-1}). \end{cases} \quad (1)$$

With  $l = 1$  and  $l = 0$  we obtain the line and immittance spectral pair polynomials, respectively. The original polynomial  $A(z)$  can be reconstructed by  $A(z) = \frac{1}{2}[P(z) + Q(z)]$ .

Given that  $A(z)$  is minimum-phase, the polynomials  $P(z)$  and  $Q(z)$  will have their roots interlaced on the unit circle. In addition, from [12] we know that the roots are separated by a positive margin.

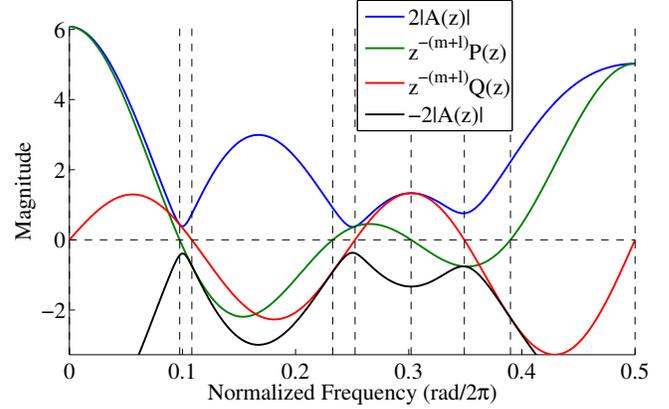
The polynomials  $P(z)$  and  $Q(z)$  are symmetric and antisymmetric, respectively, with the axis of symmetry at  $z^{-(m+l)/2}$ . It follows that the spectra of  $z^{-(m+l)/2}P(z)$  and  $z^{-(m+l)/2}Q(z)$ , respectively, evaluated on the unit circle  $z = \exp(i\theta)$  are real and complex valued. Since the zeros are on the unit circle, we can find them by searching for zero-crossings. Moreover, the evaluation on the unit-circle can be implemented efficiently using an FFT.

As the spectra corresponding to  $z^{-(m+l)/2}P(z)$  and  $z^{-(m+l)/2}Q(z)$  are real and complex, respectively, we can implement them with a single FFT. Specifically, if we take the sum  $z^{-(m+l)/2}(P(z) + Q(z))$  then the real and complex parts of the spectra correspond to  $z^{-(m+l)/2}P(z)$  and  $z^{-(m+l)/2}Q(z)$ , respectively. Moreover, since

$$z^{-(m+l)/2}(P(z) + Q(z)) = 2z^{-(m+l)/2}A(z), \quad (2)$$

we can directly take the FFT of  $2z^{-(m+l)/2}A(z)$  to obtain the spectra corresponding to  $z^{-(m+l)/2}P(z)$  and  $z^{-(m+l)/2}Q(z)$ , without explicitly determining  $P(z)$  and  $Q(z)$ . Since we are interested only in the locations of zeros, we can omit scalar multiplication and evaluate  $z^{-(m+l)/2}A(z)$  by FFT instead. Observe that since  $A(z)$  has only  $m+1$  non-zero coefficients, we can use FFT pruning to reduce complexity [13].

To ensure that all roots are found, we must use an FFT of sufficiently high length  $N$  that the spectrum is evaluated on



**Fig. 1.** Illustration of the real and complex parts of  $z^{-(m+l)/2}A(z)$ . The vertical dashed lines depict the line spectral frequencies.

at least one frequency between every two zeros. In comparison to evaluating the magnitudes  $|P(z)|$  and  $|Q(z)|$ , the zero-crossing approach has a significant advantage in accuracy. Consider, for example, the sequence 3, 2, -1, -2. With the zero-crossing approach it is obvious that the zero lies between 2 and -1. However, by studying the corresponding magnitude sequence 3, 2, 1, 2, we can only conclude that the zero lies somewhere between the second and the last elements. In other words, with the zero-crossing approach the accuracy is double in comparison to the magnitude-based approach.

Figure 1 illustrates the relation of  $A(z)$ ,  $P(z)$  and  $Q(z)$ . Note that the magnitude is expressed on a linear axis rather than on the decibel scale in order to keep zero-crossings visible. We can see that the line spectral frequencies occur at the zeros crossings of  $P(z)$  and  $Q(z)$ . Moreover, the magnitudes of  $P(z)$  and  $Q(z)$  are smaller than or equal to  $2|A(z)|$  everywhere;

$$|P(e^{i\theta})| \leq 2|A(e^{i\theta})| \quad \text{and} \quad |Q(e^{i\theta})| \leq 2|A(e^{i\theta})|. \quad (3)$$

The equality occurs only at the line spectral frequencies. This relationship can be readily derived using Eq. 1 and a continuity argument.

In summary, given a linear predictor with coefficients  $a_k$ , we can obtain the line spectral frequencies by applying the FFT of length  $N$  on the zero-extended sequence  $a_k$  and by first applying the phase-rotation  $z^{(m+l)/2}$  and then searching for zero-crossings of the real and complex parts. Since the zeros of  $P(z)$  and  $Q(z)$  are interlaced, we can alternate between searching for zeros on the real and complex parts, such that we find all zeros in one pass, and reduce complexity by half in comparison to a full search. Further, to refine the locations of zeros, we can apply interpolation in the neighborhood of the zeros.

### 3. FIXED POINT IMPLEMENTATION

Speech codecs are often implemented on mobile devices with limited resources, whereby numerical operations must be implemented with fixed-point representations. It is therefore essential that algorithms implemented operate with numerical representations whose range is limited. For common speech spectral envelopes, the numerical range of the Fourier spectrum is, however, so large that we need a 32-bit implementation of the FFT to ensure that the location of zero-crossings are retained.

A 16-bit FFT can, on the other hand, often be implemented with lower complexity, whereby it would be beneficial to limit the range of spectral values to fit within that 16-bit range. Our approach is based on pre-conditioning the spectrum of  $A(z)$  with a stabilizing filter  $B(z)$ . From Eq. 3 we see that by limiting the numerical range of  $B(z)A(z)$  we also limit the numerical range of  $B(z)P(z)$  and  $B(z)Q(z)$ . If  $B(z)$  does not have zeros on the unit circle, then  $B(z)P(z)$  and  $B(z)Q(z)$  will have the same zero-crossing on the unit circle as  $P(z)$  and  $Q(z)$ . Moreover,  $B(z)$  has to be symmetric such that  $z^{-(m+l+n)/2}P(z)B(z)$  and  $z^{-(m+l+n)/2}Q(z)B(z)$  remain symmetric and antisymmetric and their spectra are purely real and imaginary, respectively. Instead of evaluating the spectrum of  $z^{(m+l)/2}A(z)$  we can thus evaluate  $z^{(m+l+n)/2}A(z)B(z)$ , where  $B(z)$  is an order  $n$  symmetric polynomial without roots on the unit circle. In other words, we can apply the same approach as in Section 2, but first multiplying  $A(z)$  with filter  $B(z)$  and applying a modified phase-shift  $z^{-(m+l+n)/2}$ .

The remaining task is to design a filter  $B(z)$  such that the numerical range of  $A(z)B(z)$  is limited, with the restriction that  $B(z)$  must be symmetric and without roots on the unit circle. The simplest filter which fulfills the requirements is an order 2 linear-phase filter

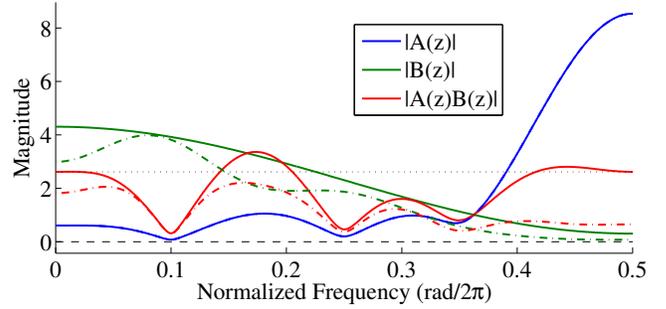
$$B_1(z) = \beta_0 + \beta_1 z^{-1} + \beta_0 z^{-2} \quad (4)$$

where  $\beta_k \in \mathbb{R}$  are the parameters and  $|\beta_1| > 2|\beta_0|$ . By adjusting  $\beta_k$  we can modify the spectral tilt and thus reduce the numerical range of the product  $A(z)B_1(z)$ . A computationally very efficient approach is to choose  $\beta_k$  such that the magnitude at 0-frequency and Nyquist is equal,  $|A(1)B_1(1)| = |A(-1)B_1(-1)|$ , whereby we can choose for example

$$\beta_0 = A(1) - A(-1) \quad \text{and} \quad \beta_1 = 2(A(1) + A(-1)). \quad (5)$$

This approach provides an approximately flat spectrum. Figure 2 illustrates the performance.

We observe that whereas  $A(z)$  has a high-pass character,  $B_1(z)$  is low-pass, whereby the product  $A(z)B_1(z)$  has, as expected, equal magnitude at 0- and the Nyquist-frequency and it is more or less flat. Since  $B_1(z)$  has only one degree of freedom, we obviously cannot expect that the product would be completely flat. Still, observe that the ratio between the



**Fig. 2.** Illustration of the magnitude spectrum of a filter  $A(z)$ , the corresponding flattening filters  $B_1(z)$  and  $B_2(z)$  and the products  $A(z)B_1(z)$  and  $A(z)B_2(z)$ , where the solid and dash-dotted lines correspond to  $B_1(z)$  and  $B_2(z)$  respectively. The horizontal dotted line shows the level of  $A(z)B_1(z)$  at the 0- and Nyquist-frequencies.

highest peak and lowest valley of  $B_1(z)A(z)$  is much smaller than that of  $A(z)$ . This means that we have obtained the desired effect; the numerical range of  $B_1(z)A(z)$  is much smaller than that of  $A(z)$ .

A second, slightly more complex method is to calculate the autocorrelation  $r_k$  of the impulse response of  $A(0.5z)$ . Here multiplication by 0.5 moves the zeros of  $A(z)$  in the direction of origin, whereby the spectral magnitude is reduced approximately by half. By applying the Levinson-Durbin on the autocorrelation  $r_k$ , we obtain a filter  $H(z)$  of order  $n$  that is minimum-phase [14]. We can then define the order  $2n$  filter  $B_2(z) = z^{-n}H(z)H(z^{-1})$  to obtain a  $|B_2(z)A(z)|$  that is approximately constant. This filter is illustrated in Fig. 2 with the dash-dotted lines. We can see that the range of  $|B_2(z)A(z)|$  is smaller than that of  $|B_1(z)A(z)|$ .

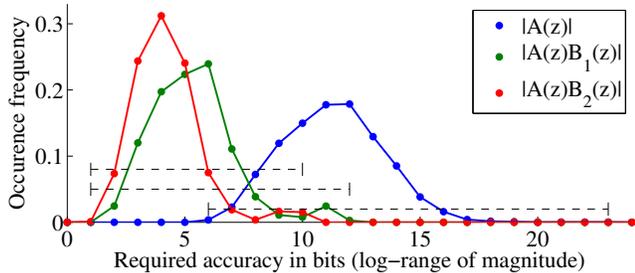
Further approaches for the design of  $B(z)$  can be readily found in classical literature of FIR design [15].

### 4. EXPERIMENTS

To verify the performance of the proposed algorithm, we performed two experiments; First, we estimated the number of bits required for the numerical representation of the spectra as well as the accuracy of the obtained line spectral frequencies. Second, we measured the computational complexity in a fully featured speech codec.

As data for all our experiments we used the set of speech, audio and mixed sound samples used in evaluation of the MPEG USAC standard [16] with a sampling rate of 12.8 kHz. The audio samples were windowed with Hamming windows of length 30 ms and pre-filtered with  $1 - 0.68z^{-1}$ .

In the first experiment, our purpose was to determine whether the method presented in Section 3 is sufficient to reduce the numerical range of the spectrum to allow usage a 16-bit implementation of the FFT. The required numerical



**Fig. 3.** Histograms of the number of required bits to represent the magnitude spectra of  $A(z)$  and  $A(z)B(z)$ , respectively. The horizontal dashed lines illustrate the range of values.

Method	Mean square error	
	$N = 200$	$N = 256$
Chebyshev	$2.51 \times 10^{-8}$	$0.91 \times 10^{-8}$
FFT $A(z)$	$2.70 \times 10^{-8}$	$1.38 \times 10^{-8}$
FFT $A(z)B_1(z)$	$2.75 \times 10^{-8}$	$1.39 \times 10^{-8}$
FFT $A(z)B_2(z)$	$1.77 \times 10^{-8}$	$1.03 \times 10^{-8}$

**Table 1.** Accuracy of line spectral frequencies with different methods and different numbers of evaluation points  $N$ .

range of the spectrum can be estimated by taking the  $\log_2$ -ratio of the highest peak and lowest valley of the spectrum, that is, for a filter  $C(z)$ , the required bits can be estimated by

$$b = \log_2 \frac{\max_{\theta} |C(\exp(i\theta))|}{\min_{\theta} |C(\exp(i\theta))|}. \quad (6)$$

For each frame of the test material, we calculated the required number of bits for the original linear predictive model  $A(z)$  and the stabilized spectra  $A(z)B_1(z)$  and  $A(z)B_2(z)$ , where  $B_1(z)$  and  $B_2(z)$  are defined as in Section 3 and  $B_2(z)$  is of order  $2m$ , where  $m = 16$  is the order of  $A(z)$ . The histograms of both approaches are depicted in Figure 3.

We can see that for the linear predictor  $A(z)$ , we need a fixed point representation with an accuracy up to 23 bits, whereas  $A(z)B_1(z)$  and  $A(z)B_2(z)$  can be represented with 12 and 10 bits, respectively. In other words, by convolving  $A(z)$  with  $B_1(z)$  or  $B_2(z)$ , the maximal numerical range of the spectrum is reduced from 23 to 12 or 10 bits. This successfully demonstrates that for  $A(z)$  a 32-bit implementation of the FFT is required, while 16-bits is sufficient for both  $A(z)B_1(z)$  and  $A(z)B_2(z)$ . Since the simple and low-complexity approach to stabilization of  $B_1(z)$  is adequate for the current application, a study of more refined approaches was not necessary.

The algorithm  $B_1(z)$  was then implemented in a full-scale speech codec, a working draft version of the 3GPP Enhanced Voice Services standard, where the order of  $A(z)$  is  $m = 16$ . The proposed method was implemented in C, with an FFT length of  $N = 256$ . As comparison, we included the search algorithm of the G.718 standard [2]. It uses a zero-crossing

Approach	Complexity (WMOPS)		
	min	max	mean
G.718	0.698	0.758	0.735
Proposed	0.352	0.393	0.377

**Table 2.** Computational complexity of the line spectrum frequency search in G.718 and the proposed approach, measured by Weighted Million Operations per Second (WMOPS).

search in the Chebyshev-domain on a grid of 100 points. This corresponds to an FFT of length 200, since the FFT includes aliased components above the Nyquist frequency. The accuracy of Chebyshev and the proposed methods are listed in Table 1. As a reference value for measuring root accuracy we used the root locations calculated by the MATLAB function `roots()`. We can see that we can reach higher accuracy than Chebyshev with  $N = 200$  by selecting either the FFT method with  $A(z)B_2(z)$  with  $N = 200$  or any of the FFT methods with  $N = 256$ . Since  $B_1(z)$  is very simple to implement, is more accurate at  $N = 256$  than the standard Chebyshev at  $N = 200$ , we chose to apply this method in the EVS standard.

The computational complexity of the G.718 and the proposed method based on FFT with  $A(z)B_1(z)$  and  $N = 256$  are listed in Table 2 as measured by Weighted Million Operations per Second (WMOPS) [17]. Most importantly, the maxima of observed WMOPS values are for G.718 and the proposed method 0.758 and 0.393, respectively. We have thus obtained a 48% reduction in computational complexity.

## 5. DISCUSSION

We present a method for finding the line spectral frequencies of linear predictors, which is applicable to speech codecs for mobile devices, such as the recent 3GPP Enhanced Voice Services standard [4]. The proposed method can be applied with higher accuracy than conventional methods, such as the search applied in G.718, while simultaneously reducing computational complexity by 48%.

Conventional methods are usually based on the Chebyshev transform, which is not orthonormal, whereby the numerical accuracy rapidly decreases with an increase in the model order. Since the proposed method is based on the Fourier transform, which is orthonormal, it can readily be applied also to higher-order predictors. The proposed method is thus applicable also to the problem of estimating the basis frequencies of Vandermonde transforms [18, 19].

In summary, the proposed method is widely applicable for finding line spectral frequencies, especially in speech coding. It has already been adopted to the 3GPP Enhanced Voice Services standard [4].

## References

- [1] B. Bessette, R. Salami, R. Lefebvre, M. Jelinek, J. Rotola-Pukkila, J. Vainio, H. Mikkola, and K. Järvinen, “The adaptive multirate wideband speech codec (AMR-WB),” *IEEE Trans. Speech Audio Process.*, vol. 10, no. 8, pp. 620–636, 2002.
- [2] ITU-T G.718, “Frame error robust narrow-band and wideband embedded variable bit-rate coding of speech and audio from 8–32 kbit/s,” 2008.
- [3] M. Neuendorf, M. Multrus, N. Rettelbach, G. Fuchs, J. Robilliard, J. Lecomte, S. Wilde, S. Bayer, S. Disch, C. Helmrich, R. Lefebvre, P. Gournay, B. Bessette, J. Lapierre, K. Kjörning, H. Purnhagen, L. Villemoes, W. Oomen, E. Schuijers, K. Kikuri, T. Chinen, T. Norimatsu, K. S. Chong, E. Oh, M. Kim, S. Quackenbush, and B. Grill, “The ISO/MPEG unified speech and audio coding standard – consistent high quality for all content types and at all bit rates,” *Journal of the AES*, vol. 61, no. 12, 2013.
- [4] 3GPP, *TS 26.445, EVS Codec Detailed Algorithmic Description; 3GPP Technical Specification (Release 12)*, 2014.
- [5] F. Itakura, “Line spectrum representation of linear predictor coefficients of speech signals,” *The Journal of the Acoustical Society of America*, vol. 57, pp. S35, 1975.
- [6] F. Soong and B. Juang, “Line spectrum pair (LSP) and speech data compression,” in *Proc. ICASSP*, 1984, vol. 9, pp. 37–40.
- [7] T. Bäckström and C. Magi, “Properties of line spectrum pair polynomials – a review,” *Signal Processing*, vol. 86, no. 11, pp. 3286–3298, Nov. 2006.
- [8] P. P. Vaidyanathan, “The theory of linear prediction,” in *Synthesis Lectures on Signal Processing*, vol. 2, pp. 1–184. Morgan & Claypool publishers, 2007.
- [9] G. Kang and L. Fransen, “Application of line-spectrum pairs to low-bit-rate speech encoders,” in *Proc. ICASSP*. IEEE, 1985, vol. 10, pp. 244–247.
- [10] P. Kabal and R. P. Ramachandran, “The computation of line spectral frequencies using Chebyshev polynomials,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 34, no. 6, pp. 1419–1426, 1986.
- [11] S. S. Yedlapalli, “Transforming real linear prediction coefficients to line spectral representations with a real FFT,” *IEEE Trans. Speech Audio Process.*, vol. 13, no. 5, pp. 733–740, 2005.
- [12] T. Bäckström, C. Magi, and P. Alku, “Minimum separation of line spectral frequencies,” *IEEE Signal Process. Lett.*, vol. 14, no. 2, pp. 145–147, Feb. 2007.
- [13] R. G. Alves, P. L. Osorio, and M. N. S. Swamy, “General FFT pruning algorithm,” in *Circuits and Systems, 2000. Proc 43rd IEEE Midwest Symposium*, 2000, vol. 3, pp. 1192–1195.
- [14] M. H. Hayes, *Statistical Digital Signal Processing and Modeling*, John Wiley & Sons, 1996.
- [15] T. Saramäki, “Finite impulse response filter design,” in *Handbook for Digital Signal Processing*, pp. 155–277. Wiley, 1993.
- [16] M. Neuendorf, P. Gournay, M. Multrus, J. Lecomte, B. Bessette, R. Geiger, S. Bayer, G. Fuchs, J. Hilpert, N. Rettelbach, R. Salami, G. Schuller, R. Lefebvre, and B. Grill, “Unified speech and audio coding scheme for high quality at low bitrates,” in *Proc. ICASSP*, 2009, pp. 1–4.
- [17] ITU-T G.191, “Software tools for speech and audio coding standardization,” 2005.
- [18] T. Bäckström, “Vandermonde factorization of Toeplitz matrices and applications in filtering and warping,” *IEEE Trans. Signal Process.*, vol. 61, no. 24, pp. 6257–6263, Dec. 2013.
- [19] T. Bäckström, J. Fischer, and D. Boley, “Implementation and evaluation of the Vandermonde transform,” in *EUSIPCO 2014*, Lisbon, Portugal, Sept. 2014.