On Pole-Zero Model Estimation Methods Minimizing a Logarithmic Criterion for Speech Analysis

Damián Marelli and Peter Balazs, *Member, IEEE*

***Abstract*—A speech production model consists of a linear, slowly time-varying filter. Pole-zero models are required for a good representation of certain types of speech sounds, like nasals and laterals. From a perceptual point of view, designing them by minimizing a logarithmic criterion appears as a very suitable approach. The most accurate available results are obtained by using Newton-like search algorithms to optimize pole and zero positions, or the coefficients of a decomposition into quadratic factors. In this paper, we propose to optimize the numerator and denominator coefficients instead. Experimental results show that this is the computationally most efficient approach, especially when the optimization criterion considers a psychoacoustical frequency scale. To illustrate its applicability in speech processing, we used the proposed method for formant and anti-formant tracking as well as speech resynthesis.**

***Index Terms*—Bark scale, estimation, iterative methods, logarithmic arithmetic, nasals, numerator and denominator, poles and zeroes, speech analysis, transfer functions.**

I. Introduction

A speech production model consists of a linear, slowly time-varying filter, called the speech production filter (SPF), whose input is a combination of a train of impulses and white noise [1]–[3]. The SPF models the combined effect of the vocal tract and the radiation at the lips, as well as the glottal pulse shape in the case of voiced sounds, and it is assumed to be time-invariant during a short-time period (frame) of approximately 20–40 ms. Applications of this technique can be found in speech coding [4], speech synthesis [5], speaker recognition [6], [7] and automatic speech recognition [8].

A well-known approach to estimate the SPF is linear predictive coding (LPC) [1], [2], [9], [10]. The LPC technique models the SPF as an all-pole linear system whose coefficients are obtained by adapting a predictor of the output signal based on its own previous samples. The use of all-pole models provides a good representation for the majority of speech sounds. However, the representation of some sounds like nasals, fricatives, laterals, or the burst interval of stop consonants, require the use of a pole-zero model [2], [4], as these kinds of sounds contain spectral zeros which are difficult to approximate with an all-pole model. Detecting zeros is a relevant issue in speaker recognition [11], automatic speech recognition [8], speech analysis and coding [12], among others.

Another approach to improve the performance of LPC analysis is called perceptual linear predictive (PLP) analysis [13], and consists in using a psychoacoustical frequency scale (e.g., the Bark scale [14]) when estimating the SPF. These frequency scales are fitted to human auditory perception and are often used in audio applications like speech coding [15] or speech recognition [16].

A pole-zero model estimation method aims to optimize the coefficients of the numerator and the denominator of the SPF to match some estimate of its frequency response. Some methods estimate the numerator and denominator separately, e.g., Prony's method [2] or those in [17], [18]. Others do it jointly [19], [20]. However, the best performances are obtained using recursive methods [21]–[26]. Motivated by the fact that the human auditory system is perceiving amplitude of the frequency contents of a sound signal in a logarithmic scale [27], some methods aim at minimizing a logarithmic criterion. An early attempt was done in [28]. The authors of [29] provide a suboptimal solution to this problem by using a recursive weighted linear least-squares (WLLS) procedure. While this algorithm offers satisfactory results with a few number of iterations, it is not guaranteed to converge. Even if it does so, it does not reach a local minimum in general. To avoid this problem, the authors of [30] optimized the positions of poles and zeros using a Newton-like algorithm. A disadvantage of this method is that the numbers of real and complex poles and zeros, as well as their multiplicity order, need to be known *a priori* (or obtained from a "guess"). The former information is not required by the method in [31] which optimizes the coefficients of the decomposition of the numerator and the denominator into quadratic factors, instead of pole and zero positions. However, this method still requires the knowledge of the multiplicity order of each quadratic factor. In another research line, and with the aim of improving the computational complexity of the method in [30], the authors of [32] approximated the optimization problem in the cepstral domain. However, a drawback of this approach is that its computational efficiency is undermined if a psychoacoustical frequency scale is used in the optimization criterion.

The purpose of this paper is of twofold. First we give a survey of the available methods for estimating a pole-zero model mini-

Manuscript received September 17, 2008; revised May 22, 2009. First published June 19, 2009; current version published November 13, 2009. This work was supported by the EU Marie Curie fellowship MEIF-CT-2006-023728 and was partially developed at the Faculty of Mathematics, NuHAG, University of Vienna. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Simon King.

D. Marelli is with the School of Electrical Engineering and Computer Science, University of Newcastle, Callaghan, NSW 2308, Australia (e-mail: damian.marelli@newcastle.edu.au).

P. Balazs is with the Acoustics Research Institute, Austrian Academy of Sciences, 1040 Vienna, Austria (e-mail: peter.balazs@oeaw.ac.at).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASL.2009.2025544

mizing a logarithmic criterion in a unified mathematical framework. In the process of doing so, we propose to carry out the optimization using a quasi-Newton algorithm, as opposite to the Newton algorithm used in [30]–[32]. Second, we propose an alternative approach to the methods in [30]–[32], in which we directly optimize the numerator and denominator coefficients in the frequency domain. By doing so, neither the numbers of real and complex poles and zeros, nor their multiplicity order, need to be known *a priori*. Also, as we show in Section V, and confirm experimentally in Section VI, the proposed approach is computationally more efficient than the methods in [30]–[32]. This computational advantage is even bigger when the optimization criterion is defined using a psychoacoustical frequency scale. A potential disadvantage of the proposed approach in comparison with those in [30]–[32] is that it does not permit a direct control of pole and zero locations during the iterative procedure, which could be advantageous in some applications. Another disadvantage is that the numerator and denominator coefficients are more sensitive to quantization errors than pole and zero positions, or the coefficients of a decomposition into quadratic factors. While this turns the proposed approach into an unattractive option for applications using fixed-point arithmetic, experimental results using real speech samples show that this is not an issue when floating-point arithmetic is used. While the motivating application considered in this work is the estimation of a model for the SPF, the proposed algorithm can also find applications in the modeling of head-related transfer functions (HRTFs), where pole-zero modeling [31], [33] and perceptual frequency scales [34] are also relevant.

The rest of the paper is organized as follows. In Section II, we give an overview of the speech production model, and in Section III, of those estimation methods which aim at minimizing a logarithmic criterion. In Section IV, we introduce the proposed estimation method. In Section V we do a computational cost analysis of all available methods and in Section VI, we evaluate their performance using real speech samples. We give concluding comments in Section VII.

II. Digital Modeling of Speech Signals

In the speech production model, the sampled speech signal $y(t)$ is assumed to be generated by an excitation signal $u(t)$ filtered by the SPF, i.e., if $g_t(\tau)$ denotes the time-variant impulse response of the SPF, then

$$y(t) = \sum_{\tau \in \mathbb{Z}} g_t(\tau) u(t-\tau). \tag{1}$$

For unvoiced sounds, the signal $u(t)$ is assumed to be white noise. In the case of voiced sounds, it is assumed to be a train of impulses, i.e.,

$$u(t) = \sum_{k \in \mathbb{Z}} \delta(t - kT_0) \tag{2}$$

where $\delta(t)$ denotes the Dirac delta function and T_0 denotes the *pitch period*.

The speech signal is divided into frames. For each frame, the SPF is assumed to be LTI and a parametric model $G(z,\theta)$ of the SPF (θ denotes the vector of parameters), is tuned according to some optimization criterion.

A. All-Pole Model Estimation

In this approach, $G(z,\theta)$ is built using an all-pole transfer function [1], [2]. More precisely, $\theta = [a_1, \ldots, a_m]^T$ (m denotes the order of the denominator) and

$$G(z,\theta) = \frac{1}{1 + a_1 z^{-1} + \cdots + a_m z^{-m}}.$$

Then, for each frame, the parameters are chosen as follows:

$$\theta = \arg\min_{\theta'} \sum_{t \in \mathcal{T}} \left| y(t) + \bar{A}(q,\theta') y(t) \right|^2$$

where $\bar{A}(q,\theta) = a_1 q^{-1} + \cdots + a_m q^{-m}$ (q denotes the forward time-shift operator, i.e., $qy(t) = y(t+1)$) and $\mathcal{T}$ denotes the time interval of the corresponding frame.

B. Pole-Zero Model Estimation

These methods model the SPF as a transfer function having zeros as well as poles. As mentioned in Section I, this is necessary for a better representation of some kind of speech sounds. Generally speaking, the available methods consist of two steps.

SPF's Frequency Response Estimation: The first step consists in estimating the frequency response of the SPF. This can be done by using the all-pole method described above with a model of very high order [35], [36], using the homomorphic prediction method [37], estimating cepstral coefficients [38], [39], etc. Each method has its advantages and it is outside the scope of this work to compare their performances. In this paper, we use the method described in [40], which builds the SPF's frequency response by interpolating spectral peaks found within neighborhoods of the multiples of the pitch frequency.

Pole-Zero Model Tuning: The second step consists in tuning a pole-zero model to fit the frequency response estimated above. The SPF is modeled as

$$G(z,\theta) = \frac{B(z,\theta)}{A(z,\theta)} \tag{3}$$

where θ is a set of parameters (e.g., containing the numerator and denominator coefficients, or the real and imaginary components of poles and zeros). This is a classical problem in nonlinear estimation theory and there is a number of available recursive and non-recursive methods. Some methods aim to solve the following optimization problem (or a weighted version of it):

$$\theta = \arg\min_{\theta'} \sum_{k=1}^{K} \left| \hat{G}(\omega_k) - \frac{B(e^{j\omega_k},\theta')}{A(e^{j\omega_k},\theta')} \right|^2 \tag{4}$$

where $\hat{G}(\omega_k)$ is the estimate of the SPF's frequency response and $\{\omega_k, k = 1, \ldots, K\}$ is a discrete set of frequencies [19]–[22], [25]. However, as mentioned in Section I, the logarithm of the amplitude of the frequency contents of a sound signal can be a more appropriate measure. Hence, a more suitable optimization criterion is [28]–[31]

$$\theta = \arg\min_{\theta'} \sum_{k=1}^{K} \left| \log \hat{G}(\omega_k) - \log \frac{B(e^{j\omega_k},\theta')}{A(e^{j\omega_k},\theta')} \right|^2. \tag{5}$$

The minimization problems in (4) and (5) require the phase information of the SPF frequency response estimate $\hat{G}(\omega_k)$. The

phase information is usually obtained from the amplitude information by assuming that the SPF has minimum-phase. This assumption is not strictly correct in a physical sense since, as mentioned in [1], the SPF is a non-minimum phase system. However, the assumption is still perceptually valid because the human auditory system is for the most part insensitive to phase information [27]. Hence, the phase information can be ignored during the optimization procedure, leading to the following optimization criterion [30], [32]

$$\theta = \arg\min_{\theta'} \sum_{k=1}^{K} \left| \log\left|\hat{G}(\omega_k)\right| - \log\left|\frac{B(e^{j\omega_k},\theta')}{A(e^{j\omega_k},\theta')}\right| \right|^2. \quad (6)$$

The advantage of using (6) over (5) is that, as mentioned in [31], it achieves a smaller amplitude error in general. Notice that the model obtained by solving (6) may contain poles and/or zeros outside the unit circle. To guarantee the stability and the minimum phase assumption described above, those poles and zeros lying outside the unit circle are reflected inside by inverting their magnitudes.

Bark Scale: As mentioned in Section I, the use of psychoacoustical frequency scales is a common practice in speech signal processing. One of such scales is the Bark scale [14] in which, for a given frequency f in Hertz, its corresponding Bark value f_b is given by

$$f_b = 13 \cdot \arctan(0.00076 \cdot f) + 3.5 \cdot \arctan\left(\frac{f}{7500}\right)^2. \quad (7)$$

The criteria (5) and (6) can be modified so that the optimization is carried out using this frequency scale. This can be done by either uniformly distributing the grid $\{\omega_k, k = 1, \ldots, K\}$ in the desired frequency scale or adding a spectral weighting function within the optimization criterion.

III. Methods for Minimizing a Logarithmic Criterion

In this section, we give an overview of the available methods for solving (5) and (6). In Section III-A, we describe a relatively simple iterative linear least-squares algorithm which, while not producing an optimal model in general, it requires few computations and is therefore an attractive option to initialize more sophisticated methods using Newton-like optimization techniques. We give a unified view of all these methods in Section III-B, and in Sections III-B1to III-B3 we summarize the different methods.

A. Estimation of Numerator and Denominator Coefficients Using WLLS

An algorithm to solve (5) was proposed in [29], which consists of an iterative procedure in which a previous estimation of the numerator and denominator coefficients is used to build a WLLS problem to jointly estimate a new set of coefficients. More precisely, the SPF is modeled as

$$G(z,\theta) = \frac{\sum_{l=0}^{n} b_l z^{-l}}{1 + \sum_{l=1}^{m} a_l z^{-l}} \quad (8)$$

where n and m denote the orders of the numerator and denominator, respectively. The set of parameters θ is chosen as

$$\theta = [b_0, b_1, \ldots, b_n, a_1, \ldots, a_m]^T \quad (9)$$

and its value θ_i at iteration i is given by

$$\theta_i = \arg\min_{\theta'} \sum_{k=1}^{K} W_i(e^{j\omega}) \left| \hat{G}(\omega_k) A(e^{j\omega_k},\theta') - B(e^{j\omega_k},\theta') \right| \quad (10)$$

with $W_0(e^{j\omega}) = 1$ and

$$W_i(e^{j\omega}) = \left| \frac{\log \hat{G}(\omega_k) - \log B(e^{j\omega_k},\theta_{i-1}) + \log A(e^{j\omega_k},\theta_{i-1})}{\hat{G}(\omega_k) A(e^{j\omega_k},\theta_{i-1}) - B(e^{j\omega_k},\theta_{i-1})} \right|^2. \quad (11)$$

An algorithm similar to (10) and (11), but used for solving (4) instead of (5), was given in [21], [22]. It was shown in [25] that this algorithm, even when it converges, it does not necessarily converge to a local minimum of the desired cost function. To avoid this problem, an improvement of this algorithm was proposed [25, Eq. 14]. It was pointed out in [41] that this algorithm is equivalent to applying the Gauss–Newton algorithm (without linear search) to (4). Hence, by following an analysis similar to the one in [25], it follows that the algorithm (10), (11) does not converge to a local minimum of (5) in general, and that a way to avoid this problem is to use a Newton-like search method. Nevertheless, this algorithm yields satisfactory results and, as reported by the authors, requires few iterations to converge.

B. Estimation Using Newton-Like Search Methods

Equations (5) and (6) are nonlinear least-squares problems which, as done in [30]–[32], can be solved using any Newton-like search algorithm. We give a unifying view of these algorithms below. For a comprehensive presentation see [42].

The optimization problems (5) and (6) can be written as

$$\theta = \arg\min_{\theta'} V(\theta') \quad (12)$$

$$V(\theta) = \sum_{k=1}^{K} [F(\theta)]_k^2 \quad (13)$$

where $[F(\theta)]_k$ denotes the kth component of the real-valued vector $F(\theta)$, which is a function of the d-dimensional real-valued vector θ. Then, (12), (13) are equivalent to (5) if we define $F(\theta) = [F_m(\theta), F_p(\theta)]^T$ with

$$[F_m(\theta)]_k = \log\left|\frac{\hat{G}(\omega_k)}{G(e^{j\omega_k},\theta)}\right| \quad (14)$$

$$[F_p(\theta)]_k = \angle\frac{\hat{G}(\omega_k)}{G(e^{j\omega_k},\theta)} \quad (15)$$

for all $k = 1, \ldots, K$ ($\angle x$ denotes the angle of x), and equivalent to (6) if we define $F(\theta) = F_m(\theta)$. Using Newton-like methods, (12), (13) is solved using the following iterative procedure:

$$\theta_{i+1} = \theta_i - \alpha_i \tilde{\theta}_i \tag{16}$$

where $\tilde{\theta}_i$ is the solution of

$$H_i \tilde{\theta}_i = g_i \tag{17}$$

the scalar α_i denotes the step size at iteration i, the d-dimensional vector g_i denotes the gradient of $V(\theta)$ at θ_i, and the $d \times d$ matrix H_i denotes either the Hessian of $V(\theta)$ at θ_i or an approximation of it.

Let $J(\theta)$ denote the Jacobian of $F(\theta)$, i.e.,

$$[J(\theta)]_{k,l} = \frac{\partial [F(\theta)]_k}{\partial [\theta]_l}. \tag{18}$$

The gradient g_i can be computed from the Jacobian information by

$$g_i = 2J^T(\theta_i) F(\theta_i). \tag{19}$$

The different Newton-like methods differ in the way in which H_i is defined. The Newton method, which was considered in [30]–[32], consists of taking H_i as the Hessian matrix of $V(\theta)$ at θ_i. As pointed out in [42], from a practical point of view this method has some drawbacks. First, it requires the implementation effort and complexity associated with the computation of the second order derivatives of $V(\theta)$ at θ_i. Second, the Hessian H_i may not be positive definite if θ_i is remote from a local minimum, and therefore some kind of correction needs to be introduced (e.g., replacing H_i by $H_i + \lambda I$, where λ is chosen to make $H_i + \lambda I$ positive definite). Third, it requires the solution of a system of d linear equations at each iteration. In this paper, we consider the quasi-Newton method, which avoids these drawbacks by directly approximating H_i^{-1} using some iterative procedure. A popular choice is the Broyden–Fletcher–Goldfarb–Shanno (BFGS) formula which is given by [42]

$$\begin{aligned} H_{i+1}^{-1} &= H_i^{-1} + \left(1 + \frac{q_i^T H_i^{-1} q_i}{s_i^T q_i}\right) \frac{s_i s_i^T}{s_i^T q_i} \\ &\quad - \frac{s_i q_i^T H_i^{-1} + H_i^{-1} q_i s_i^T}{s_i^T q_i} \\ s_i &= \theta_{i+1} - \theta_i \\ q_i &= g_{i+1} - g_i. \end{aligned}$$

A third Newton-like option is the Gauss-Newton method, in which H_i is calculated using the following approximation of the Hessian matrix

$$H_i = 2J^T(\theta_i) J(\theta_i). \tag{20}$$

The advantage of the Gauss-Newton method is that it requires fewer iterations than quasi-Newton methods in general. However, this advantage is somehow counterbalanced by the fact that it still requires the solution of a system of d linear equations at each iteration, and the computation of (20) can be costly if a large number K of frequency points is considered. Besides this, the main drawback of this approach for our considered application is that (20) is a good approximation of the Hessian matrix of $V(\theta)$ only if the residual error $\min_\theta V(\theta)$ is small. This is not the case in general in our context, and consequently the method may fail to converge. Therefore, we do not consider this method in our application.

Finally, the step-size parameter α_i is obtained from a linear search algorithm. In this paper, following [30] and [31], we implement it by using a sub-iterative procedure (i.e., formed of *sub-iterations* of the *main iterations* (16), (17)) in which, starting from the initial value $\alpha_i = 1$, the value of α_i is halved at each sub-iteration until

$$V(\theta_i - \alpha_i \tilde{\theta}_i) < V(\theta_i)$$

or a maximum number of iterations is reached.

Different approaches have been proposed to solve (5) and (6) using Newton-like methods. They differ on the way in which $G(z, \theta)$ is parametrized. We summarize them next.

1) Estimation of Poles and Zeros (PZ): In [30], $G(z, \theta)$ is parametrized using its poles and zeros. More precisely, $G(z, \theta)$ is written as

$$G(z, \theta) = C \frac{\prod_{l=1}^{n} (1 - \beta_l z^{-1})}{\prod_{l=1}^{m} (1 - \alpha_l z^{-1})}$$

and the vector of parameters θ in (3) contains the gain constant C as well as the real and imaginary components of α_l and β_l. The optimization problems (5) and (6) can then be written in the form (12), (13), and the resulting first order derivatives required to build the Jacobian matrix (18) are given by

$$\frac{\partial [F_m(\theta)]_k}{\partial C} = -\frac{1}{C}$$

for the gain constant

$$\frac{\partial [F_m(\theta)]_k}{\partial \beta_l} = \Re \left\{ \frac{r_l e^{-j\omega_k}}{1 - \beta_l e^{-j\omega_k}} \right\} \tag{21}$$

for a real zero of multiplicity r_l, and

$$\frac{\partial [F_m(\theta)]_k}{\partial \Re\{\beta_l\}} = \Re \left\{ \frac{r_l e^{-j\omega_k}}{1 - \beta_l e^{-j\omega_k}} + \frac{r_l e^{-j\omega_k}}{1 - \overline{\beta_l} e^{-j\omega_k}} \right\} \tag{22}$$

$$\frac{\partial [F_m(\theta)]_k}{\partial \Im\{\beta_l\}} = \Re \left\{ \frac{j r_l e^{-j\omega_k}}{1 - \beta_l e^{-j\omega_k}} - \frac{j r_l e^{-j\omega_k}}{1 - \overline{\beta_l} e^{-j\omega_k}} \right\} \tag{23}$$

for a pair of complex conjugate zeros of multiplicity r_l. The derivatives with respect to pole positions are obtained by taking the negative values of the right-hand side of (21)–(23) and replacing β_l by α_l. Also, the derivatives of F_p are obtained by taking the imaginary components instead of the real components in (21)–(23).

2) Estimation of Quadratic Factors (QF): Suppose we assume that all poles and zeros of $G(z, \theta)$ have multiplicity 1. Even under this assumption, in order to build the vector of parameters θ for the method in Section III-B1, the user still needs to know the number of real and complex poles and zeros (this information is obtained from the initialization), which will remain

unchanged during the iterative procedure. In [31], this limitation is avoided by expressing (3) in quadratic factors, i.e.,

$$G(z,\theta) = C\frac{\prod_{l=1}^{n/2}(1 + b_{1,l}e^{-j\omega_k} + b_{2,l}e^{-2j\omega_k})}{\prod_{l=1}^{m/2}(1 + a_{1,l}e^{-j\omega_k} + a_{2,l}e^{-2j\omega_k})}$$

and defining θ to contain the gain constant C as well as the quadratic factors' parameters $a_{1,l}$, $a_{2,l}$, $l = 1,\ldots,m/2$, and $b_{1,l}$, $b_{2,l}$, $l = 1,\ldots,n/2$. The first-order derivatives are given by

$$\frac{\partial\,[F_m(\theta)]_k}{\partial b_{1,l}} = \Re\left\{\frac{-r_l e^{-j\omega_k}}{1 + b_{1,l}e^{-j\omega_k} + b_{2,l}e^{-2j\omega_k}}\right\} \quad (24)$$

$$\frac{\partial\,[F_m(\theta)]_k}{\partial b_{2,l}} = \Re\left\{\frac{-r_l e^{-j2\omega_k}}{1 + b_{1,l}e^{-j\omega_k} + b_{2,l}e^{-2j\omega_k}}\right\} \quad (25)$$

where r_l denotes the multiplicity of the quadratic factor. As before, the derivatives with respect to $a_{1,l}$ and $a_{2,l}$ are obtained by taking the negative values of (24) and (25) and replacing b by a; and the derivatives of F_p use the imaginary components instead of the real components in (24) and (25).

3) Estimation of Poles and Zeros in the Cepstral Domain (PZ-Ceps): Aiming to simplify the Jacobian expression (21)–(23) when solving (6), the authors of [32] approximated the problem of Section III-B-1 in the cepstral domain. More precisely, $F(\theta)$ is replaced by

$$[\mathcal{C}_{F_m}(\theta)]_1 = \mathcal{C}_{\hat{G}}(0) - 2\log C \quad (26)$$

$$[\mathcal{C}_{F_m}(\theta)]_{k+1} = \mathcal{C}_{\hat{G}}(k) - \frac{1}{k}\left(\sum_{i=1}^{n_D}\alpha_i^k - \sum_{i=1}^{n_N}\beta_i^k\right) \quad (27)$$

for all $k = 1,\ldots,K-1$, where $\mathcal{C}_{\hat{G}}(k)$ denotes the inverse Fourier transform of $\log|\hat{G}(\omega_k)|^2$. Notice that C is directly obtained from (26), so only α and β need to be estimated. The resulting first-order derivatives are given by

$$\frac{\partial\,[\mathcal{C}_{F_m}(\theta)]_k}{\partial\beta_l} = \beta_l^{k-1} \quad (28)$$

for a real zero of multiplicity r_l, and

$$\frac{\partial\,[\mathcal{C}_{F_m}(\theta)]_k}{\partial\Re\{\beta_l\}} = 2r_l\Re\left\{\beta_l^{k-1}\right\} \quad (29)$$

$$\frac{\partial\,[\mathcal{C}_{F_m}(\theta)]_k}{\partial\Im\{\beta_l\}} = -2r_l\Im\left\{\beta_l^{k-1}\right\} \quad (30)$$

for a pair of complex conjugate zeros of multiplicity r_l. Again, the derivatives with respect to pole positions are obtained by taking the negative values of (28)–(30) and replacing β_l by α_l.

While this method is computationally more efficient than the method in Section III-B1 (see Section V), its efficiency is undermined if a psychoacoustical frequency scale is used in the optimization criterion (6). To see this, recall that considering such a scale is equivalent to consider a linear frequency scale together with a frequency weighting function in (6). This in turn implies that the right-hand side of (26), (27) is affected by a convolution, which complicates the first derivative expressions (28)–(30).

Remark 1: Notice that the gain constant C can be explicitly computed using (26), and hence removed from the vector θ of estimated parameters, even when using the methods in Sections III-B1 and III-B2 to solve either (5) or (6).

IV. Proposed Approach: Estimation of Numerator and Denominator Coefficients (ND)

The advantage of estimating quadratic factors instead of pole and zeros positions is that the number of real and complex poles, as well as zeros, need not be know *a priori*. However, this approach still requires the a priori knowledge of the multiplicity order of each quadratic factor. The natural approach to go around this limitation is to directly optimize the numerator and denominator coefficients. A potential drawback of this approach is that these coefficients are sensitive to quantization error [43, Ch. 7.6]; hence, it is not an attractive option in implementations using fixed point arithmetic. However, as shown in Section VI, this is not an issue if floating point arithmetic is used. Additionally, and more importantly, this approach offers computational advantages compared to the methods in [30]–[32], as shown below.

Define $G(z,\theta)$ and θ as in (8) and (9), respectively. It is straightforward to verify that the first order derivatives are given by

$$\frac{\partial\,[F_m(\theta)]_k}{\partial b_l} = \Re\left\{\frac{-e^{-jl\omega_k}}{B(e^{j\omega_k},\theta)}\right\} \quad (31)$$

$$\frac{\partial\,[F_m(\theta)]_k}{\partial a_l} = \Re\left\{\frac{e^{-jl\omega_k}}{A(e^{j\omega_k},\theta)}\right\}. \quad (32)$$

Again, the derivatives of F_p are obtained by taking the imaginary components instead of the real components in (31) and (32).

The computational advantages of this approach over the algorithms in [30]–[32] come from the following two facts.

1) It permits an efficient implementation of the linear search algorithm. To see this, notice that

$$\left[F(\theta_n - \alpha\tilde{\theta}_n)\right]_k = \log\left|\frac{\hat{G}(\omega_k)A(e^{j\omega_k},\theta_n) + \alpha\hat{G}(\omega_k)\bar{A}(e^{j\omega_k},\tilde{\theta}_n)}{B(e^{j\omega_k},\theta_n) + \alpha B(e^{j\omega_k},\tilde{\theta}_n)}\right| \quad (33)$$

with $\bar{A}(z,\theta) = A(z,\theta) - 1 = a_1z^{-1} + \cdots + a_mz^{-m}$. The terms $\hat{G}(\omega_k)B(e^{j\omega_k},\theta_n)$, $\hat{G}(\omega_k)\bar{A}(e^{j\omega_k},\tilde{\theta}_n)$, $B(e^{j\omega_k},\theta_n)$, and $B(e^{j\omega_k},\tilde{\theta}_n)$, for all $\omega_k : k = 1,\ldots,K$, need only be computed once for the whole line search, simplifying in this way its associated computational cost.

2) Notice that the kth row $[J(\theta)]_{k,:}$ of the Jacobian matrix can be written as

$$[J(\theta)]_{k,:} = \Re\left[-\frac{W_B(\omega_k)}{B(e^{j\omega_k},\theta)}, \frac{W_A(\omega_k)}{A(e^{j\omega_k},\theta)}\right]$$

where $W_B(\omega) = [1, e^{-j\omega},\ldots,e^{-j\omega n}]$ and $W_A(\omega) = [e^{-j\omega},\ldots,e^{-j\omega m}]$. So (19) can be efficiently computed by using

$$F^T(\theta)J(\theta) = \Re\{-\Phi_B(\theta)\Omega_B, \Phi_A(\theta)\Omega_A\} \quad (34)$$

with

$$\Phi_B(\theta) = \left[\frac{[F(\theta)]_1}{B(e^{j\omega_1},\theta)},\ldots,\frac{[F(\theta)]_K}{B(e^{j\omega_K},\theta)}\right] \quad (35)$$

$$\Omega_B = \left[W_B^T(\omega_1),\ldots,W_B^T(\omega_K)\right]^T \quad (36)$$

and $\Phi_A(\theta)$ and Ω_A defined by replacing B by A in (35) and (36). Notice that if the frequency points $\omega_k : k = 1,\ldots,K$ are equally spaced, Ω_B and Ω_A are $K \times K$ discrete Fourier transform (DFT) matrices with missing columns. Hence, further computational savings can be achieved using a fast Fourier transform (FFT) algorithm to compute (34).

Remark 2: It is often convenient to express an iterative algorithm in a compact form as in (10) and (11). If the Gauss–Newton method is used to solve (6), as shown in Appendix A, the proposed method can be written as follows:

$$\theta_i = \arg\min_{\theta} \sum_{k=1}^{K} \Re\left\{\left|\alpha_i \log\frac{\hat{G}(\omega_k)A(e^{j\omega_k},\theta_{i-1})}{B(e^{j\omega_k},\theta_{i-1})} + \frac{B(e^{j\omega_k},\theta)}{B(e^{j\omega_k},\theta_{i-1})} - \frac{A(e^{j\omega_k},\theta)}{A(e^{j\omega_k},\theta_{i-1})}\right\}\right|^2. \quad (37)$$

To solve (5), the operation $\Re\{\cdot\}$ has to be removed from (37).

V. Computational Cost Analysis

In this section, we carry out a computational analysis of the different methods when solving (6), which is a preferred criterion in speech processing. We consider the WLLS method of Section III-A, the method in Section III-B1 which optimizes pole and zero positions (denoted by PZ), is cepstral domain counterpart (PZ-Ceps) described in Section III-B3, the method in Section III-B2 which estimates quadratic factors (QF), the proposed approach estimating the numerator and denominator coefficients (ND), and its variant using FFT (ND-FFT) to compute (34) when a regular grid of frequency point is used in (6).

We use the number of real multiplications as the performance index for comparison purposes. More precisely, a real division counts as one real multiplication, a complex multiplication requires four real multiplications and a complex division requires eight. If K is a power of two, a K-point FFT computed using the Radix-2 algorithm requires $2K\log_2 K$ multiplications [43, Ch. 6.1]. The computation of the logarithm of a real positive number can be done with ten multiplications. To see this, we write the natural logarithm of a real positive number $x = m2^n$, with mantissa $1 \le m < 2$, as $\ln(x) = \ln(2^{-0.5}m) + (n + 0.5)\ln(2)$. Then we use $\ln(2^{-0.5}m) \simeq 2\sum_{i=0}^{3}(1/(2i+1))((2^{-0.5}m - 1)/(2^{-0.5}m+1))^{2i+1}$ which guarantees a relative error smaller than 10^{-7} in the whole range $1 \le m < 2$. Finally, the solution of the system of $2K$ equations with d unknowns, corresponding to each iteration (10), requires approximately $2d^2K$ multiplications [44, Ch. 5.3].

Let $d = m + n - 1$ denote the total number of parameters to be estimated, and suppose that the number of frequency points K is a power of two. In Table I we show, for each method, the approximate number of real multiplications required for each main iteration and each sub-iteration.

TABLE I
Approximate Number of Real Multiplications Required by Each Main and Sub-Iteration of the Different Methods

	Main iteration	Sub-iteration
WLLS	$2d^2K$	-
PZ	$4d^2 + 13dK$	$(27 + 6d)K$
PZ-Ceps	$4d^2 + 2dK$	dK
QF	$4d^2 + 6dK$	$(27 + 4d)K$
ND	$4d^2 + (12 + 2d)K$	$20K$
ND-FFT	$4d^2 + (12 + 4\log_2 K)K$	$20K$

VI. Experimental Results

The proposed pole-zero model estimation method is both, locally optimal (in the sense of providing a local minimum of the desired error measure) and computationally efficient. In Sections VI-A and VI-B, we provide experimental results confirming this two properties. Also, in order to illustrate the performance of the proposed method in speech processing applications, we use it for formant and anti-formant tracking in Section VI-D as well as for speech resynthesis in Section VI-E.

In the experiments below, we use speech recordings from a female Icelandic speaker, a female Italian speaker, a male Albanian speaker, a male German speaker, and a male speaker from the Viennese dialect. The speech recording is done in a sound booth [semi anechoic chamber (IAC 1202A)], using an AKG CK91 microphone, a DAT Recorder Tascam with DAT Tapes Fuji 64p, a sampling rate of 44.1 kHz and 16 bits per sample. Before processing, the speech signals are resampled at a frequency which is specified below, since it depends on the experiment being carried out. Each speech signal is split in frames of 48 ms, with a time-shift of 10 ms between frames, and each frame is multiplied by a Hanning window to improve the shape of spectral peaks. The envelope of each frame is then obtained using the method described in [40].

For comparative purposes, we consider the method PZ (Section III-B1), which is replaced by its cepstral domain counterpart PZ-Ceps (Section III-B3) when a linear frequency scale is used in (6). We also consider the method QF (Section III-B2), and the proposed ND method (Section IV), which is replaced by its variant ND-FFT in the case of a linear frequency scale. We also include three non-optimized methods (i.e., without involving a on Newton-like search). The first is the WLLS method described in Section III-A, the second is the iterative bootstrapping LPC (IBLPC) method introduced in [24], and the third is the iterative prefiltering (IPF) method considered in [3], [23], which aims at minimizing (4). The outcome of optimized methods is highly dependent on the method used for its initialization. We initialize them using the WLLS method, since as shown in Tables II–V, this option leads to the most accurate results in general. As in [30] and [31], these iterative algorithms are set to stop when

$$\frac{e_{\log}(\theta_n) - e_{\log}(\theta_{n-1})}{e_{\log}(\theta_{n-1})} < 10^{-7}. \quad (38)$$

As explained in Section II-B, to guarantee stability and minimum phase, the estimated poles and zeros having absolute values bigger than one are reflected inside the unit circle by

TABLE II
AVERAGE PERFORMANCE OF THE DIFFERENT METHODS OVER 100 SPEECH SAMPLES USING LINEAR FREQUENCY SCALE, $m = 20$ AND $n = 10$

	Av. value of $e_{\log}$	Variance of $e_{\log}$	Av. number of main iterations	Av. number of sub-iterations	Av. number of multiplications
IPF	10.27×10^{-2}	20.54×10^{-4}	5.6	0	13×10^6
IBLPC	8.974×10^{-2}	13.60×10^{-4}	3	0	6.3×10^6
WLLS	5.595×10^{-2}	3.994×10^{-4}	3.4	0	8.6×10^6
PZ-Ceps	4.749×10^{-2}	2.788×10^{-4}	74	498	21×10^6
QF	4.738×10^{-2}	2.776×10^{-4}	69	468	86×10^6
ND-FFT	4.698×10^{-2}	2.713×10^{-4}	52	508	13×10^6

TABLE III
AVERAGE PERFORMANCE OF THE DIFFERENT METHODS OVER 100 SPEECH SAMPLES USING LINEAR FREQUENCY SCALE, $m = 20$ AND $n = 20$

	Av. value of $e_{\log}$	Variance of $e_{\log}$	Av. number of main iterations	Av. number of sub-iterations	Av. number of multiplications
IPF	7.798×10^{-2}	16.29×10^{-4}	5.5	0	22.5×10^6
IBLPC	7.080×10^{-2}	9.863×10^{-4}	6	0	20×10^6
WLLS	3.723×10^{-2}	2.752×10^{-4}	4.7	0	19.7×10^6
PZ-Ceps	3.201×10^{-2}	1.720×10^{-4}	105	657	37×10^6
QF	3.214×10^{-2}	1.736×10^{-4}	101	618	147×10^6
ND-FFT	3.173×10^{-2}	1.672×10^{-4}	64	707	18×10^6

TABLE IV
AVERAGE PERFORMANCE OF THE DIFFERENT METHODS OVER 100 SPEECH SAMPLES USING BARK FREQUENCY SCALE, $m = 20$ AND $n = 10$

	Av. value of $e_{\log}$	Variance of $e_{\log}$	Av. number of main iterations	Av. number of sub-iterations	Av. number of multiplications
IPF	11.84×10^{-2}	36.94×10^{-4}	5	0	11.9×10^6
IBLPC	11.21×10^{-2}	26.15×10^{-4}	5	0	10.5×10^6
WLLS	6.816×10^{-2}	10.00×10^{-4}	2.7	0	7.3×10^6
PZ	5.630×10^{-2}	6.892×10^{-4}	84	504	145×10^6
QF	5.709×10^{-2}	6.750×10^{-4}	77	472	88×10^6
ND	5.563×10^{-2}	6.000×10^{-4}	57	520	15×10^6

TABLE V
AVERAGE PERFORMANCE OF THE DIFFERENT METHODS OVER 100 SPEECH SAMPLES USING BARK FREQUENCY SCALE, $m = 20$ AND $n = 20$

	Av. value of $e_{\log}$	Variance of $e_{\log}$	Av. number of main iterations	Av. number of sub-iterations	Av. number of multiplications
IPF	8.440×10^{-2}	15.20×10^{-4}	5	0	20.7×10^6
IBLPC	8.063×10^{-2}	11.29×10^{-4}	6	0	20×10^6
WLLS	4.272×10^{-2}	3.512×10^{-4}	3.3	0	15×10^6
PZ	3.600×10^{-2}	1.969×10^{-4}	108	647	241×10^6
QF	3.580×10^{-2}	1.884×10^{-4}	106	608	146×10^6
ND	3.535×10^{-2}	1.797×10^{-4}	68	715	22×10^6

inverting their magnitudes. We use floating-point, double-precision (64 bits) arithmetic, and to quantify the residual estimation error we use

$$e_{\log}(\theta) = \frac{1}{K}\sum_{k=1}^{K}\left|\log\left|\hat{G}(e^{j\omega_k})\right| - \log\left|\frac{B(e^{j\omega_k},\theta)}{A(e^{j\omega_k},\theta)}\right|\right|^2.$$

A. Computational Efficiency

In this section, we evaluate the complexity of the different methods for estimating a pole-zero model for the SPF using the design criterion (6). To this end we used a resampling frequency of $f_s = 16$ kHz. We have chosen a set of 100 sound segments each containing a phoneme with a spectral zero (anti-formant), and we have obtained a frame from the center of each segment. We have considered the following phonemes and allophones:

- voiced bilabial nasal consonants /m/;
- voiced alveolar nasal consonants /n/;
- voiced velar nasal consonants /η/;
- voiced apical alveolar lateral consonants /l/;
- voiced apical dental lateral consonants with retracted tongue body /l/ [45];
- voiced apical alveolar monolateral consonants with retracted tongue body, which we denote by l^V/.[1]

An issue in the estimation of a pole-zero model is the selection of the numerator and denominator orders (i.e., n and m). Following the rule of thumb $m = f_s/1000 + 4$ [1] we have chosen $m = 20$. Unfortunately, there is no standard choice for n; however, it is reasonable to state that 20 is an upper bound for

[1]This consonant appears in the Viennese Dialect. It is produced with an apical closure at the alveolar ridge. The tongue body is retracted and lowered, and the pharynx is narrowed. The air escapes only from one side of the tongue.

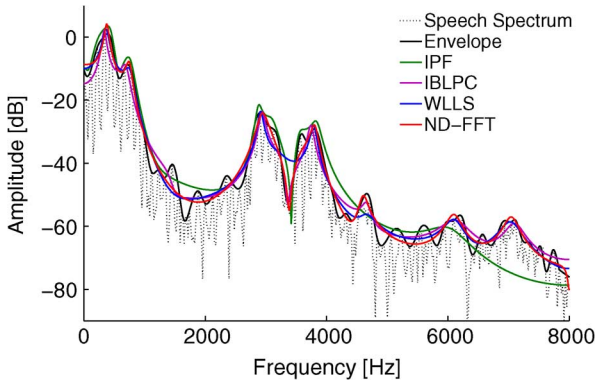

Fig. 1. Estimation of a monolateral consonant /l^V/ using IPF ($e_{\log} = 11.73 \times 10^{-2}$), IBLPC ($e_{\log} = 6.764 \times 10^{-2}$), WLLS ($e_{\log} = 6.738 \times 10^{-2}$), and ND-FFT ($e_{\log} = 5.571 \times 10^{-2}$), with $m = 20$ and $n = 10$.

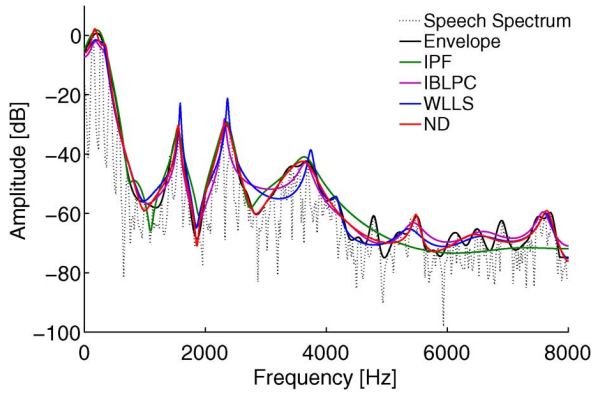

Fig. 2. Estimation of a lateral consonant /l/ usingIPF ($e_{\log} = 12.97 \times 10^{-2}$), IBLPC ($e_{\log} = 10.41 \times 10^{-2}$), WLLS ($e_{\log} = 10.67 \times 10^{-2}$), and ND-FFT ($e_{\log} = 8.369 \times 10^{-2}$), with $m = 20$ and $n = 10$.

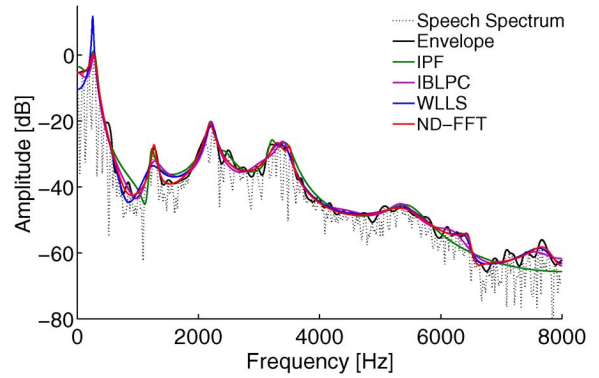

Fig. 3. Estimation of a lateral consonant /l/ using IPF ($e_{\log} = 9.216 \times 10^{-2}$), IBLPC ($e_{\log} = 6.429 \times 10^{-2}$), WLLS ($e_{\log} = 6.479 \times 10^{-2}$), and ND-FFT ($e_{\log} = 4.591 \times 10^{-2}$), with $m = 20$ and $n = 10$.

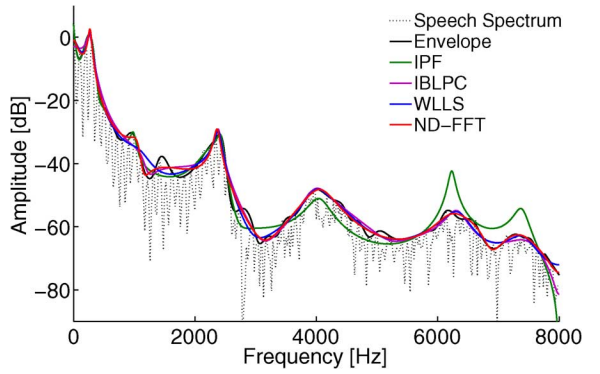

Fig. 4. Estimation of a nasal consonant /n/ using IPF ($e_{\log} = 11.51 \times 10^{-2}$), IBLPC ($e_{\log} = 5.405 \times 10^{-2}$), WLLS ($e_{\log} = 4.955 \times 10^{-2}$), and ND-FFT ($e_{\log} = 3.932 \times 10^{-2}$), with $m = 20$ and $n = 10$.

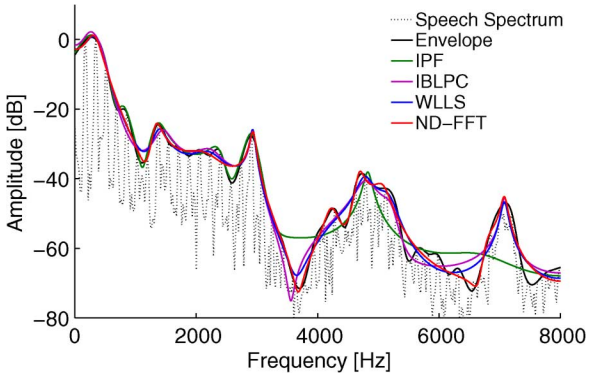

Fig. 5. Estimation of a nasal consonant /n/ using IPF ($e_{\log} = 15.45 \times 10^{-2}$), IBLPC ($e_{\log} = 9.304 \times 10^{-2}$), WLLS ($e_{\log} = 7.617 \times 10^{-2}$), and ND-FFT ($e_{\log} = 5.672 \times 10^{-2}$), with $m = 20$ and $n = 10$.

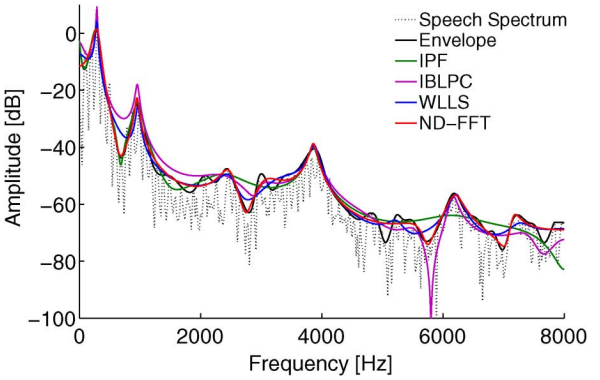

Fig. 6. Estimation of a nasal consonant /m/ using IPF ($e_{\log} = 11.68 \times 10^{-2}$), IBLPC ($e_{\log} = 12.86 \times 10^{-2}$), WLLS ($e_{\log} = 7.418 \times 10^{-2}$), and ND-FFT ($e_{\log} = 5.063 \times 10^{-2}$), with $m = 20$ and $n = 10$.

any choice. Hence, we have carried out the experiments using two values, $n = 10$ and $n = 20$.

In the first experiment, we evaluate the performance of the methods IPF, IBLPC, WLLS, PZ-Ceps, QF, and ND-FFT, when using a linear frequency scale. To this end we use $K = 1024$ regularly spaced frequency points ω_k, $k = 1, \ldots, K$ in the range $[0, f_s/2]$. We show in Tables II and III the results obtained using both, $n = 10$ and $n = 20$. We see that the errors obtained using the PZ-Ceps, QF, and ND-FFT methods are similar. Also, the PZ-Ceps and ND-FFT methods are clearly more efficient than the QF method, with ND-FFT requiring nearly half the computations required by the PZ-Ceps method.

In the second experiment, we evaluate the performance of the methods IPF, IBLPC, WLLS, PZ, QF, and ND when using a nonlinear frequency scale. We use $K = 1024$ frequency points in the range $[0, f_s/2]$, regularly spaced in the Bark scale. We show in Tables IV and V the results obtained using both, $n = 10$ and $n = 20$. Again, we see that the errors obtained using the

PZ, QF, and ND methods are similar; and in this case, the ND method is clearly the most efficient option.

B. Local Optimality

Tables II–V show that the methods IPF, IBLPC, and WLLS have a complexity comparable to that of the proposed methods ND and ND-FFT. However, the former do not yield a local optimum of 6 in general, and in some cases can lead to inaccurate estimates. To illustrate this point, we show in Figs. 1–6, the outcomes of the methods IPF, IBLPC, WLLS, and ND-FFT. We see that, while the WLLS method produces a model which approximately follows the speech envelope, it can be affected by inaccuracies, which in Fig. 1 result in the model not properly reproducing the spectral zero at about 3.3 KHz, and in Figs. 2 and 3 having resonant frequencies. Also, the IPF method approximately follows the speech envelope, but it is not accurate at frequencies where the amplitude of the envelope becomes negligible in the linear scale. This effect is clearly seen in Figs. 4 and 5. Finally, some examples where the IBLPC method leads to inaccurate estimates are shown in Figs. 5 and 6.

C. Using Nonlinear Frequency Scales

We see in Fig. 4 that the ND-FFT method does not reproduce the formant at about 1500 Hz, since its amplitude is not significant in the logarithmic scale. However, it may be argued that this spectral feature is more relevant than others found at higher frequencies. While this issue can be dealt with by increasing the model order, this may not be a preferred option in applications where complexity and/or number of parameters are restricted. An alternative approach is to distribute the grid of frequency points ω_k, $k = 1,\ldots,K$ so that the desired frequencies are given more importance. To illustrate this approach, we compare in Fig. 7 the outcomes of the ND-FFT method, which distributes the points ω_k, $k = 1,\ldots,K$ uniformly in the linear frequency scale, with that of the ND method using the Bark scale instead. We see that the ND method accurately follows the spectral envelope at low frequencies, including the formant at about 1500 Hz, with the price of not reproducing the zero at about 6800 Hz. A similar situation occurs with the zero at about 2600 Hz in Fig. 5. Again, we see in Fig. 8 that this zero is more accurately modeled by the ND method when using the bark frequency scale, with the price of not reproducing the zero at 4400 Hz.[2]

D. Formant and Anti-formant Tracking

In order to illustrate the applicability of the proposed method for speech analysis, we use the estimated SPF models to track the evolution of formants and anti-formants from the Italian word “capanna” (hut), pronounced by a female native speaker.

[2]Fig. 5 shows that the IPF method is also able to model the zero at about 2600 Hz, even when a linear frequency scale is used. The reason for this is that the IPF method considers a linear amplitude scale [i.e., it aims at minimizing (4)], where the magnitude of this spectral feature becomes more significant. However, as mentioned before, a drawback of this approach is that it does not permit modeling relevant spectral features which are negligible in the linear amplitude scale. This behavior can be observed at frequencies above 3500 Hz in Fig. 5.

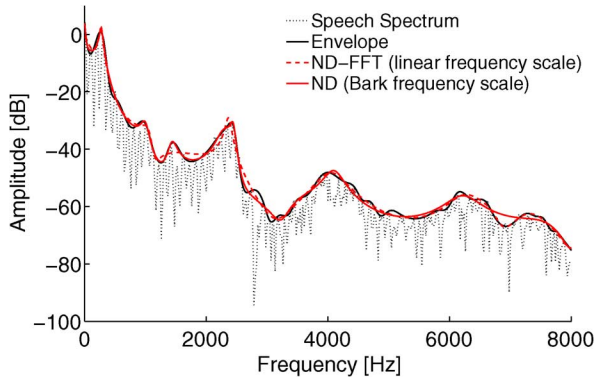

Fig. 7. Estimation of a nasal consonant /n/ using linear and Bark frequency scales, with $m = 20$ and $n = 10$.

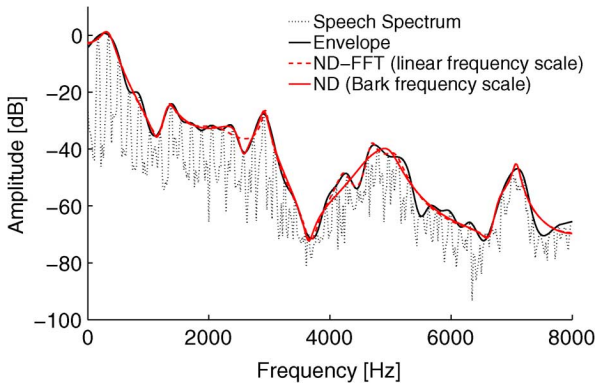

Fig. 8. Estimation of a nasal consonant /n/ using linear and Bark frequency scales, with $m = 20$ and $n = 10$.

We used a resampling frequency of $f_s = 8$ kHz, and the ND-FFT method with $n = m = 10$, optimized using a grid of $K = 1024$ frequency points regularly spaced in the linear scale. In Fig. 9, we show the poles and zeros of the estimated models, and in Fig. 10 we show the model obtained at 450 ms, corresponding to the nasal consonant /n/, which spans from approximately 350 to 500 ms. It can be seen that, during this phoneme, the first two formants are tracked by poles at about 200 and 750 Hz, being consistent with the low-frequency description of nasal consonants given in [46]. The first anti-formant is tracked by a zero at about 1200 Hz, and corresponds to the resonance of the mouth cavity closed by the tongue tip. As pointed out in [46], this anti-formant is above the second formant, and another anti-formant is expected at about three times this frequency. This is tracked in Fig. 9 by a somehow irregular string of zeros at about 3600 Hz. The third and fourth formants are tracked by poles at about 1700 and 3000 Hz, respectively. Two extra string of zeros appear at about 750 and 2200 Hz, the first of which is close to the second formant, and therefore reduces its amplitude (as seen in Fig. 10), and the second is present along the whole word. These zeros could correspond to the resonance of sinuses which, as explained in [11], are subject to individual variations and a general estimate about their location cannot be given. In any case, Fig. 9 shows the consistency of the results obtained from frames corresponding

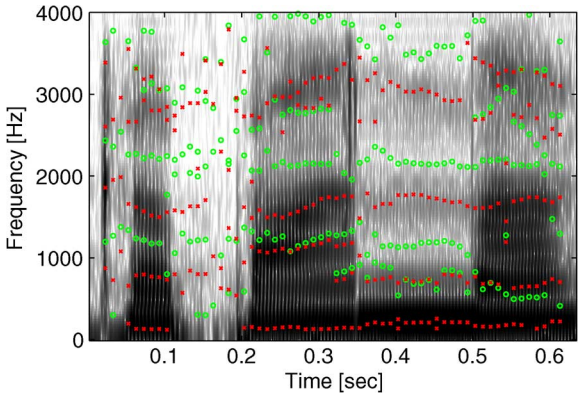

Fig. 9. Poles (x) and zeros (o) of the SPF models estimated from the word "capanna."

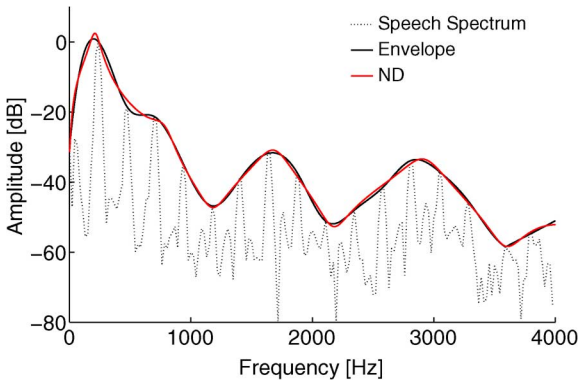

Fig. 10. Spectral slice from Fig. 9 at 450 ms. Nasal consonant /n/ in the word "capanna."

to the same phonetic element, as well as some agreement with the values expected from speech production theory.

E. Speech Resynthesis Performance

In order to illustrate the perceptual relevance of the proposed method, we used the SPF models estimated from a speech signal to resynthesize it. We used a resampling frequency of $f_s = 16$ kHz, and the ND-FFT method with $n = m = 20$, optimized using a grid of $K = 1024$ frequency points regularly spaced in the linear scale. For pitch frequency and voicing detection, we used the implementation of the RAPT algorithm [47], which forms part of the publicly available VOICEBOX Matlab toolbox. We generated the synthetic speech signal by overlap-adding speech frames, as described in [3]. We resynthesized the three German words "dümmer, dünner, dünger" (sillier, thinner, fertilizer) pronounced by a male Viennese speaker. These German words form minimal pairs in the sense that they only differ from each other in the nasal consonants /m/, /n/, and /ŋ/, and hence they can be better recognized if spectral zeros are properly reproduced. The spectograms of the original and resynthesized speech signals are shown in Figs. 11 and 12, respectively, from where we can see that they reasonably resemble each other. For a perceptual evaluation, the sound examples can be found at http://www.kfs.oeaw.ac.at/polezero.

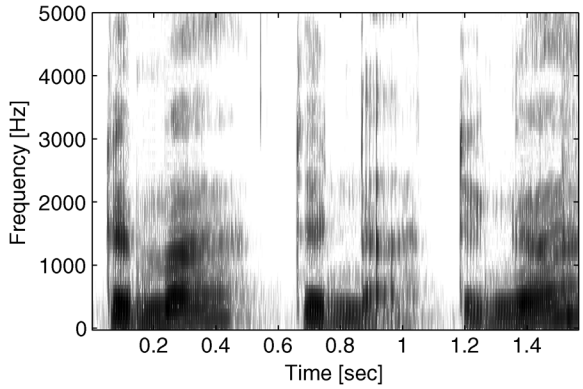

Fig. 11. Spectrograms of the original speech signal "dümmer, dünner, dünger."

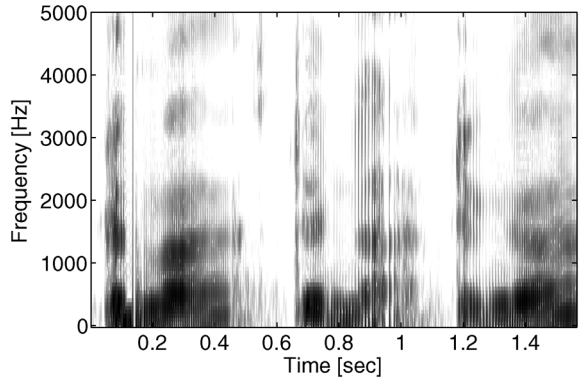

Fig. 12. Spectograms of the resynthesized speech signal "dümmer, dünner, dünger."

VII. Conclusion

We proposed a method to estimate a pole-zero model for speech production which is both, numerically efficient and optimal when minimizing a logarithm criterion. To this end, we optimize the numerator and denominator coefficients, instead of pole and zero locations, or the coefficients of a decomposition of the model into quadratic factors. We presented experimental results showing the efficiency and optimality of the proposed method. To illustrate its applicability in speech processing, we used it for formant and anti-formant tracking as well as speech resynthesis.

Appendix A
Proof of (37)

Let $J_m(\theta)$ denote the Jacobian of $F_m(\theta)$ at θ. The ith iteration of the Gauss–Newton method can be written as

$$\begin{aligned}\theta_i &= \alpha_i \arg\min_{\tilde{\theta}} \left\| F_m(\theta_{i-1}) - J_m(\theta_{i-1})\tilde{\theta} \right\|_2^2 + \theta_{i-1} \\ &= \arg\min_{\theta} \|\alpha_i F_m(\theta_{i-1}) - J_m(\theta_{i-1})(\theta - \theta_{i-1})\|_2^2 \quad (39)\end{aligned}$$

where $\|X\| = \sum_{k=1}^{K} |[X]_k|^2$. Recall that $\bar{A}(z,\theta) = A(z,\theta) - 1 = a_1 z^{-1} + \cdots + a_m z^{-m}$, then

$$[J_m(\theta_{i-1})(\theta - \theta_{i-1})]_k$$

$$= \Re\left\{ -\frac{B(e^{j\omega_k},\theta) - B(e^{j\omega_k},\theta_{i-1})}{B(e^{j\omega_k},\theta_{i-1})} + \frac{\bar{A}(e^{j\omega_k},\theta) - \bar{A}(e^{j\omega_k},\theta_{i-1})}{A(e^{j\omega_k},\theta_{i-1})} \right\}$$

$$= \Re\left\{ -\frac{B(e^{j\omega_k},\theta)}{B(e^{j\omega_k},\theta_{i-1})} + \frac{A(e^{j\omega_k},\theta)}{A(e^{j\omega_k},\theta_{i-1})} \right\} \quad (40)$$

and (37) follows from (39), (14), and (40).

ACKNOWLEDGMENT

The authors would like to thank W. A. Deutsch, S. Moosmüller, and T. Becker for discussions and ideas leading to this work, as well as M. Goupell for proofreading. The authors would also like to thank the anonymous reviewers for their valuable contributions.

REFERENCES

[1] L. Rabiner and R. Schafer, *Digital Processing of Speech Signals*. Englewood Cliffs, NJ: Prentice-Hall, 1978.

[2] J. Markel and J. A. Gray, *Linear Prediction of Speech*. Berlin, Heidelberg, Germany: Springer-Verlag, 1976.

[3] T. F. Quatieri, *Discrete-Time Speech Signal Processing: Principles and Practice (Prentice Hall Signal Processing Series)*. Upper Saddle River, NJ: Prentice-Hall , 2001.

[4] G. Fant, *Acoustic Theory of Speech Production*. The Hague, The Netherlands: Mouton, 1960.

[5] E. Keller, *Fundamentals of Speech Synthesis and Speech Recognition: Basic Concepts, State of the Art and Future Challenges*. Chichester, U.K.: Wiley, 1984.

[6] P. Rose, *Forensic Speaker Identification*. New York: Taylor & Francis, 2002.

[7] F. Nolan, *The Phonetic Bases of Speaker Recognition*. Cambridge, U.K.: Cambridge Univ. Press, 1983.

[8] M. Benzeghiba, R. D. Mori, O. Deroo, S. Dupont, T. Erbes, D. Jouvet, L. Fissore, P. Laface, A. Mertins, C. Ris, R. Rose, V. Tyagi, and C. Wellekens, "Automatic speech recognition and speech variability: A review," *Speech Commun.*, vol. 49, no. 10–11, pp. 763–786, 2007.

[9] B. Atal and M. Schroeder, "Predictive coding of speech signals," in *Proc. 6th lnt. Congr. Acoust.*, 1968, pp. 13–16.

[10] F. Itakura and S. Saito, "Analysis synthesis telephony based on the maximum likelihood method," in *Proc. 6th Int. Congr. Acoust.*, 1968, pp. 17–20.

[11] K. Johnson, *Acoustic and Auditory Phonetics*. Oxford, U.K.: Blackwell, 1997.

[12] T. van Waterschoot and M. Moonen, "Linear prediction of audio signals," in *Proc. Interspeech*, Antwerp, Belgium, Aug. 2007, pp. 518–521.

[13] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *J. Acoust. Soc. Amer.* vol. 87, no. 4, pp. 1738–1752, 1990 [Online]. Available: http://link.aip.org/link/?JAS/87/1738/1

[14] E. Zwicker and E. Terhardt, "Analytical expressions for critical-band rate and critical bandwidth as a function of frequency," *J. Acoust. Soc. Amer.*, vol. 68, no. 5, pp. 1523–1525, Nov. 1980.

[15] R. McAulay and T. Quatieri, "Low-rate speech coding based on the sinusoidal model," in *Advances in Speech Signal Processing*, S. Furui and M. Sondhi, Eds. New York: Marcel Dekker, 1991, ch. 6, pp. 165–208.

[16] L. R. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[17] J. Shanks, "Recursion filters for digital processing," *Geophysics*, vol. 32, pp. 33–51, Feb. 1967.

[18] J. Makhoul, "Linear prediction: A tutorial review," *Proc. IEEE*, vol. 63, no. 4, pp. 561–580, 1975.

[19] R. Kalman, "Design of self-optimizing control system," *Amer. Soc. Mech. Eng.-Papers*, p. 10, 1957.

[20] E. Levi, "Complex-curve fitting," *IRE Trans. Autom. Control*, vol. 4, pp. 37–44, 1959.

[21] C. Sanathanan and J. Koerner, "Transfer function synthesis as a ratio of two complex polynomials," *IEEE Trans. Autom. Control*, vol. 8, no. 1, pp. 56–58, Jan. 1963.

[22] K. Steiglitz and L. McBride, "A technique for the identification of linear systems," *IEEE Trans. Autom. Control*, vol. AC-10, no. 10, pp. 461–464, Oct. 1965.

[23] K. Steiglitz, "On the simultaneous estimation of poles and zeros in speech analysis," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-25, no. 3, pp. 229–234, Jun. 1977.

[24] C. Schmid, "Design of IIR/FIR filters using a frequency domain bootstrapping technique and LPC methods," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-31, no. 4, pp. 999–1006, Aug. 1983.

[25] A. Whitfield, "Asymptotic behaviour of transfer function synthesis methods," *Int. J. Control*, vol. 45, no. 3, pp. 1083–1092, 1987.

[26] K. Schnell and A. Lacroix, "Pole zero estimation from speech signals by an iterative procedure," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP'01)*, 2001, vol. 1, pp. 109–112.

[27] W. Hartmann, *Signals, Sounds, and Sensation*. New York: Springer, 1998.

[28] J. Olive, "Automatic formant tracking by a Newton–Raphson technique," *J. Acoust. Soc. Amer.*, vol. 50, no. 2, pp. 661–670, 1971.

[29] T. Kobayashi and S. Imai, "Design of IIR digital filters with arbitrary log magnitude function by WLS techniques," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 38, no. 2, pp. 247–252, Feb. 1990.

[30] M. Blommer and G. Wakefield, "On the design of pole-zero approximations using a logarithmic error measure," *IEEE Trans. Signal Process.*, vol. 42, no. 11, pp. 3245–3248, 1994.

[31] M. Blommer and G. Wakefield, "Pole-zero approximations for head-related transfer functions using a logarithmic error criterion," *IEEE Trans. Speech Audio Process.*, vol. 5, no. 3, pp. 278–287, May 1997.

[32] J. Derby, "Comments on "on the design of pole-zero approximations using a logarithmic error measure"," *IEEE Trans. Signal Process.*, vol. 44, no. 7, pp. 1811–1113, Jul. 1996.

[33] A. Kulkarni and H. Colburn, "Infinite-impulse-response models of the head-related transfer function," *J. Acoust. Soc. Amer.*, vol. 115, no. 4, pp. 1714–1728, 2004.

[34] J. Huopaniemi, N. Zacharov, and M. Karjalainen, "Objective and subjective evaluation of head-related transfer function filter design," *J. Audio Eng. Soc.*, vol. 47, no. 4, pp. 218–239, 1999.

[35] K. Song and C. Un, "Pole-zero modeling of speech based on high-order pole model fitting and decomposition method," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-31, no. 6, pp. 1556–1565, Dec. 1983.

[36] P. Broersen, "Accurate ARMA models with Durbin's second method," in *Proc. ICASSP, IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1999, pp. 1597–1600.

[37] G. Kopec, A. Oppenheim, and J. Tribolet, "Speech analysis by homomorphic prediction," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 25, no. 1, pp. 40–49, Feb. 1977.

[38] T. Galas and X. Rodet, "An Improved Cepstral Method for Deconvolution of Source-Filter Systems With Discrete Spectra: Application to Musical Sound Signals," in *Proc. ICMC*, Glasgow, U.K., 1990, pp. 82–84.

[39] O. Cappe and E. Moulines, "Regularization techniques for discrete cepstrum estimation," *IEEE Signal Process. Lett.*, vol. 3, no. 4, pp. 100–102, Apr. 1996.

[40] H. Hermansky, H. Fujisaki, and Y. Sato, "Spectral envelope sampling and interpolation in linear predictive analysis of speech," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1984, vol. 9, pp. 53–56.

[41] R. Pintelon, P. Guillaume, Y. R. J. Schoukens, and H. V. Hamme, "Parametric identification of transfer functions in the frequency domain—A survey," *IEEE Trans. Autom. Control*, vol. 39, no. 11, pp. 2245–2260, Nov. 1994.

[42] R. Fletcher, *Practical Methods of Optimization*, ser. A Wiley-Interscience Publication, 2nd ed. Chichester, U.K.: Wiley, 1987.

[43] J. Proakis and D. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[44] G. Golub and C. V. Loan, *Matrix Computations*, 3rd ed. Baltimore, MD: Johns Hopkins Univ. Press, 1996.

[45] P. Ladefoged and I. Maddieson, *The Sounds of the World's Languages*. Oxford, U.K.: Blackwell, 1996.

[46] K. N. Stevens, *Acoustic Phonetics*. Cambridge, MA: MIT Press, 1999.

[47] D. Talkin, "A robust algorithm for pitch tracking (RAPT)," *Speech Coding Synth.*, pp. 495–518, 1995.

Damián Marelli received the B.S. degree in electronics engineering from the Universidad Nacional de Rosario, Rosario, Argentina, in 1995, and the Ph.D. degree in electrical engineering and the B.S. (honors) degree in mathematics from the University of Newcastle, Callaghan, Australia, in 2003.

In 2003, he held a research associate position at the School of Electrical Engineering and Computer Science, University of Newcastle. In 2004 and 2005, he held a postdoctoral research fellowship at the Laboratoire d'Analyse Topologie et Probabilités, CNRS/Université de Provence, France. Since 2006 he has been a Research Academic at the ARC Centre for Complex Dynamic Systems and Control, University of Newcastle. He held an Intra-European Marie Curie Fellowship at the Faculty of Mathematics, University of Vienna, Vienna, Austria, from 2007 to 2008. His main research interests include multirate signal processing, time–frequency analysis, system identification, and statistical signal processing.

Peter Balazs (S'02–M'05) was born in Tulln, Austria, in 1970. He received the M.S. and the Ph.D. degrees (both with distinction) in mathematics from the University of Vienna, Vienna, Austria, in 2001 and 2005, respectively.

He has been a member of the Acoustics Research Institute (ARI), Austrian Academy of Science, Vienna, since 1999. He was a fellow of the HASSIP EU network, where he collaborated with the Numerical Harmonic Analysis Group (NuHAG) from the University of Vienna, the Laboratoire d'Analyse Topologie et Probabilités (LATP-CNRS), Marseille, and the Laboratoire de Mécanique et d'Acoustique (LMA-CNRS), from 2003 to 2006, and with the Unité de physique théorique et de physique mathématique (FYMA) from the Université Catholique de LouvainLouvain-La-Neuve in 2005. He is the leader and founder of the group Mathematics and Acoustical Signal Processing, at the ARI, and has recently obtained a WWTF research project entitled "MULAC: Frame Multiplers: Theory and Application in Acoustics." This "high potential" project allowed the group to grow up to five researchers, and is framed in a collaboration with the groups LMA, LATP, FYMA, and NuHAG. He is interested in frame theory, signal processing, time–frequency analysis, masking models, Gabor analysis, numerical analysis, speaker recognition, acoustics, and psychoacoustics.

Dr. Balazs is a member of the Audio Engineering Society (AES) and the Austrian (OMG) as well as the European Mathematical Society (EMS).