REVISITING THE LINEAR PREDICTION ANALYSIS-BY-SYNTHESIS SPEECH CODING PARADIGM USING REAL-TIME CONVEX OPTIMIZATION

Daniele Giacobello[1]*, Manohar Murthi*[2]*, Tobias Lindstrøm Jensen*[3]*, Mads Græsbøll Christensen*[4]

[1]Sonos Inc., Santa Barbara, CA
[2]Electrical & Computer Engineering Department, University of Miami, FL
[3]Signal and Information Processing, Department of Electronic Systems, Aalborg University, Denmark
[4]Audio Analysis Lab, AD:MT, Aalborg University, Denmark

giacobello@ieee.org

ABSTRACT

In this work, we propose a novel approach to speech coding by rewriting the nonlinear analysis-by-synthesis linear prediction scheme as a convex problem. This allows for determining trade-offs between, on one hand, the reconstruction error and, on the other, the sparsity of the predictor and the residual used to parametrize the speech signal. Differently from traditional coding schemes where the parameters are chosen throughout multiple optimization stages, our scheme produces a one-shot parametrization of a speech segment that intrinsically takes into consideration the voiced or unvoiced nature of a speech segment providing a better balance between residual and predictor and, consequently, a more appropriate bit allocation.

Index Terms— Sparse linear prediction, convex optimization, real-time implementation, speech coding.

1. INTRODUCTION

The linear prediction analysis-by-synthesis (LPAS) paradigm has set the standard for speech coding for the past thirty years together with his most successful embodiment: code-excited linear prediction (CELP) [1,2]. In simple terms, the approach is to first find the linear prediction (LP) parameters in an open-loop configuration then searching for the best excitation that models the prediction residual given certain constraints on it. This second step is done in a closed-loop configuration where the perceptually weighted distortion between the original and synthesized speech segment is minimized. Since the predictor is quantized *transparently*[1], all the responsibility for the distortion falls on the coding of the residual. A consequence of this approach can be seen in today modern codec, e.g., the AMR-WB coder in its 23.85 kbit/s configuration, allocates 90% of the bits for the residual and only 10% for the predictor [4].

Ways to improve the LPAS suboptimal multi-step approach has been a subject of study since the early days of speech coding. In [5], a search in the set of quantized linear predictors (in the LSF domain) follows the quantization of the residual to reduce the mismatch between original and reconstructed speech. This procedure can then applied then also to the residual and iterated several times, as done in [6]. Other examples include [7] and [8] where methods to re-optimize jointly residual and predictor are introduced. In our earlier work, we proposed a way to improve performances of the LPAS loop by re-estimating the predictor *after* the quantization of the residual [9]. Hence, the predictor is not seen as a vocal tract model or as a whitening filter anymore but rather as a IIR representation of the *true* truncated impulse response that generates the reconstructed signal without distortion in the LPAS equations. In [10, 11], we also outlined the use of high-order sparse predictor as an efficient way to model a signal spectrum by using only few nonzero samples. Furthermore, the idea of synergistic ways to encode a speech signal by providing a sparse residual, rather than a minimum-variance one, has shown the effectiveness of older, and less computationally intensive, coding techniques like multi-pulse code excitation [12].

In this work, we generalized these idea by proposing to quantize predictor and residual only after determining a proper trade-off between the complexity of the residual and the complexity of the predictor, as measured by the number of nonzero coefficients to code, i.e., their cardinality or sparsity. In particular, working through the LPAS equations, we are able to linearize this NP-hard problem and formulate it as a convex optimization problem where the accuracy of the reconstruction and the sparse representation of the residual and a sparse representation of the high-order sparse predictor all appear in the same set of equations.

The paper is organized as follows. In Section 2, we give

The work of D. Giacobello was supported by the Marie Curie EST-SIGNAL Fellowship under Contract MEST-CT-2005-021175 and was carried out at the Department of Electronic Systems, Aalborg University. The work of T. L. Jensen was supported by The Danish Council for Strategic Research under grant number 4005-00122.

[1]I.e., when the two versions of coded speech, obtained using the unquantized and quantized predictor, are indistinguishable through listening [3].

a brief overview of the traditional formulation for LPAS on which most of modern speech codecs are built upon. In Section 3, we show how we modified the nonlinear LPAS problem to fit into a linear optimization problem which can be solved with convex tools. In Section 4, we show a fast convex method to solve our LPAS formulation. In Section 5, we provide experimental results that show the effectiveness of our formulation in providing more balanced bit allocations. Finally, Section 6 concludes our work.

2. COMMON FORMULATION

Let the input speech signal be partitioned into frames of length N, $\mathbf{x} = [x_0 \dots x_{N-1}]^T$. The first step in the LPAS formulation is to estimate a linear predictor. The second step is determining the excitation, usually provided by two *codebooks*: an adaptive one, responsible for the periodic pitch contribution, and a fixed one, responsible for the Gaussian-like part of the excitation [1].

2.1. Linear Prediction

Linear prediction is based on the following speech production model, where a speech sample $x(n)$ is written as a linear combination of P past samples:

$$x(n) = \sum_{p=1}^{P} a_p x(n-p) + r(n), \tag{1}$$

where $\{a_p\}$ are the prediction coefficients and $r(n)$ is the prediction error or *residual*. We consider the optimization problem associated with finding the prediction coefficient vector $\mathbf{a} \in \mathbb{R}^P$ from a set of observed real samples $x(n)$ for $n = 1, \dots, N$ so that the prediction error is minimized. Rewriting the speech production model for a segment of N speech samples $x(n)$, for $n = 1, \dots, N$, in matrix form:

$$\mathbf{x} = \mathbf{X}\mathbf{a} + \mathbf{r}, \tag{2}$$

the problem becomes:

$$\mathbf{a} = \underset{\mathbf{a}}{\operatorname{argmin}} \|\mathbf{x} - \mathbf{X}\mathbf{a}\|_q^q + \gamma \|\mathbf{a}\|_k^k, \tag{3}$$

where:

$$\mathbf{x} = \begin{bmatrix} x(N_1) \\ \vdots \\ x(N_2) \end{bmatrix}, \mathbf{X} = \begin{bmatrix} x(N_1-1) & \cdots & x(N_1-P) \\ \vdots & & \vdots \\ x(N_2-1) & \cdots & x(N_2-P) \end{bmatrix}.$$

We will consider the case $N_1 = 1$ and $N_2 = N+P$, which for $p = 2$ is equivalent to the autocorrelation method. Traditional formulation impose $\gamma = 0$, and $q = 2$. In this case, a closed form solution exist:

$$\hat{\mathbf{a}} = \underset{\mathbf{a}}{\operatorname{argmin}} \|\mathbf{x} - \mathbf{X}\mathbf{a}\|_2^2 = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{x}, \tag{4}$$

where $\mathbf{R} = \mathbf{X}^T\mathbf{X}$ is the autocorrelation matrix.

As part of a traditional speech coder, further processing is usually applied to the solution $\hat{\mathbf{a}}$, like bandwidth expansion, to avoid the general shortcomings of traditional linear prediction [13]. The predictor is the quantized using, e.g., line spectral frequencies [1].

2.2. Codebook Search

Considering the following speech synthesis equations [1]:

$$\mathbf{x}_k = \mathbf{H}_k \begin{bmatrix} \hat{\mathbf{r}}_{k-1} \\ \mathbf{r}_k \end{bmatrix} = \begin{bmatrix} \mathbf{H}_k^{\mathrm{U}} & \mathbf{H}_k^{\mathrm{L}} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{r}}_{k-1} \\ \mathbf{r}_k \end{bmatrix}, \tag{5}$$

where $\mathbf{H}_k = \begin{bmatrix} \mathbf{H}_k^{\mathrm{U}} & \mathbf{H}_k^{\mathrm{L}} \end{bmatrix}$ is the $N \times 2N$ convolution matrix obtained with the truncated impulse response of $\mathbf{a}$ which can be decomposed into a upper-triangular and lower-triangular $N \times N$ matrices, $\mathbf{H}_k^{\mathrm{U}}$ and $\mathbf{H}_k^{\mathrm{L}}$, respectively. Using these definitions, we can rewrite (5) as:

$$\mathbf{x}_k = \mathbf{H}_k^{\mathrm{L}}\mathbf{r}_k + \mathbf{H}_k^{\mathrm{U}}\hat{\mathbf{r}}_{k-1}, \tag{6}$$

where the term $\mathbf{H}_k^U\hat{\mathbf{r}}_{k-1}$ is known as the *zero input response* (ZIR) and is the quantized contribution of the previous frame. This is subtracted from the actual signal to quantize, $\mathbf{x}_k$, in order to obtain what's known as the *target signal*, which is given by

$$\tilde{\mathbf{x}}_k = \mathbf{H}_k^{\mathrm{L}}\mathbf{r}_k. \tag{7}$$

In the traditional LPAS formulation, the excitation signal at the input of the short term LP synthesis filter is constructed by adding two excitation vectors from adaptive and fixed codebooks. The speech is synthesized by feeding the two properly chosen vectors from these codebooks through the short term synthesis filter. The best excitation sequence in a codebook is chosen using an analysis-by-synthesis search procedure (from which LPAS gets its name) in which the error between the original and synthesized speech is minimized according to a perceptually weighted distortion measure, i.e.,

$$\|\mathbf{W}\left(\tilde{\mathbf{x}}_k - \mathbf{H}_k^{\mathrm{L}}\left(g_{\mathrm{a}}\mathbf{c}_k^{\mathrm{a}} + g_{\mathrm{f}}\mathbf{c}_k^{\mathrm{f}}\right)\right)\|_2^2. \tag{8}$$

The adaptive codebook takes into account the long-term redundancies, behaving similarly to a pitch predictor. Generally, this contribution is calculated first. Its contribution is then removed from the target signal and the updated target signal $\bar{\mathbf{x}}_k$ is used in the fixed codebook search:

$$\begin{aligned} &\underset{\mathbf{c}_k, g_{\mathrm{f}}}{\operatorname{argmin}} \|\bar{\mathbf{x}}_k - g_{\mathrm{f}}\mathbf{H}_k^{\mathrm{L}}\mathbf{c}_k^{\mathrm{f}}\|_2 \\ &= \underset{\mathbf{c}_k, g_{\mathrm{f}}}{\operatorname{argmin}} -2g_{\mathrm{f}}\bar{\mathbf{x}}_k^T\mathbf{H}_k^{\mathrm{L}}\mathbf{c}_k + g_{\mathrm{f}}^2\|\mathbf{H}_k^{\mathrm{L}}\mathbf{c}_k^{\mathrm{f}}\|^2 \end{aligned} \tag{9}$$

Clearly, obtaining both gains and codebook entries in (9) is a combinatorial problem, thus the general approach is to determine one variable at the time. A multitude of methods exist for this task (see, e.g., [2] and reference therein). Finally, the reconstructed speech will be:

$$\hat{\mathbf{x}}_k = \mathbf{H}_k^{\mathrm{L}}\left(g^{\mathrm{a}}\mathbf{c}_k^{\mathrm{a}} + g^{\mathrm{f}}\mathbf{c}_k^{\mathrm{f}}\right) + \mathbf{H}_k^{\mathrm{U}}\hat{\mathbf{r}}_{k-1}. \tag{10}$$

3. ALTERNATIVE FORMULATION

We can rewrite the speech synthesis equation in (5) in order to include the *analysis* matrix rather than the synthesis matrix [14–16]. Thus, (5) becomes:

$$\begin{aligned}\mathbf{r}_k &= \mathbf{A}_k \begin{bmatrix} \hat{\mathbf{x}}_{k-1} \\ \mathbf{x}_k \end{bmatrix} = \begin{bmatrix} \mathbf{A}_k^{\mathrm{U}} & \mathbf{A}_k^{\mathrm{L}} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{x}}_{k-1} \\ \mathbf{x}_k \end{bmatrix} = \\ &= \mathbf{A}_k^{\mathrm{L}} \mathbf{x}_k + \mathbf{A}_k^{\mathrm{U}} \hat{\mathbf{x}}_{k-1} \end{aligned} \tag{11}$$

where $\mathbf{A}_k$ is the $N \times 2N$ analysis matrix and can be decomposed also into the upper and lower triangular square matrices

$$\mathbf{A}_k^{\mathrm{U}} = \begin{bmatrix} 0 & a_P & \cdots & \cdots & a_2 & a_1 \\ 0 & 0 & a_P & \cdots & \cdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 & a_P \\ 0 & 0 & \cdots & \cdots & 0 & 0 \end{bmatrix},$$

and

$$\mathbf{A}_k^{\mathrm{L}} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ a_1 & 1 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ a_P & a_{P-1} & \cdots & \cdots & a_1 & 1 \end{bmatrix},$$

with $P = N - 1$. Given the structure of $\mathbf{A}_k$, we can rewrite (11) as:

$$\bar{\mathbf{X}}_k \begin{bmatrix} 1 \\ \mathbf{a}_k \end{bmatrix} = \mathbf{r}_k, \tag{12}$$

where

$$\bar{\mathbf{X}}_k = \begin{bmatrix} \check{\mathbf{x}}_k | \check{\mathbf{X}}_k \end{bmatrix} = \left[\begin{array}{c|cccc} x_{k,0} & \hat{x}_{k-1,N} & \cdots & \cdots & \hat{x}_{k-1,1} \\ x_{k,1} & \vdots & \ddots & \ddots & \hat{x}_{k-1,2} \\ & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & x_{k,0} & \hat{x}_{k-1,N} \\ x_{k,N} & x_{k,N-1} & \cdots & x_{k,1} & x_{k,0} \end{array}\right].$$

and $\mathbf{a}_k = [a_1, \ldots, a_P]^T$. We can also see that $\check{\mathbf{x}}_k = \mathbf{x}_k$. The variables $\mathbf{a}_k$ and $\mathbf{r}_k$ now appear linearly.

We can now write the LPAS optimization problem as follows:

$$\begin{array}{ll} \text{minimize} & \|\mathbf{r}_k - \check{\mathbf{x}}_k - \check{\mathbf{X}}_k \mathbf{a}_k\|_2^2, \\ \text{subject to} & \|\mathbf{a}_k\|_1 \leq \delta, \\ & \|\mathbf{r}_k\|_1 \leq \gamma, \end{array} \tag{13}$$

where we can easily control the tradeoff between the quality of the reconstruction and the sparsity of the representation via the constraints. Once a solution is found, we can reconstruct the speech segment using (11):

$$\hat{\mathbf{x}}_k = \left(\hat{\mathbf{A}}_k^{\mathrm{L}}\right)^{-1} \left(\hat{\mathbf{r}}_k - \hat{\mathbf{A}}_k^{\mathrm{U}} \hat{\mathbf{x}}_{k-1}\right). \tag{14}$$

Note that if we solve the unconstrained formulation of (13), we will obtain the trivial solution $\mathbf{r} = \check{\mathbf{x}}_k = \mathbf{x}_k$ and $\mathbf{a} = \mathbf{0}$. The choice of γ is then critical, as the trivial solution also fulfills the constraint $\|\mathbf{a}\|_1 \leq \delta, \forall \delta \geq 0$.

3.1. Analogies with Operational Rate-Distortion Theory

From a more general perspective, the problem in LPAS coders is to minimize the distortion $D(\cdot)$, usually in a perceptual domain, between the original signal $\mathbf{x}$ and its synthesized version $\hat{\mathbf{x}}$ subject to some constraints regarding the rate (omitting k for clarity):

$$\begin{array}{ll} \text{minimize} & D(\mathbf{x}, \hat{\mathbf{x}}), \\ \text{subject to} & R(\hat{\mathbf{x}}) \leq R^*, \end{array} \tag{15}$$

where R^* is the maximum rate allowed. This is known as operational rate distortion [17] and applications of it can be found throughout the literature, notably [18, 19].

Considering the synthesis matrix $\mathbf{H}$ used in the LPAS equations as a nonlinear transformation of $\mathbf{a}$:

$$\mathbf{H} = \mathbf{\Phi}(\mathbf{a}), \tag{16}$$

with $\mathbf{\Phi}(\cdot)$ being the nonlinear operator that maps $\mathbf{a}$ into the matrix $\mathbf{H}$, we can then write the distortion term as:

$$D(\mathbf{x}, \hat{\mathbf{x}}) = \|\mathbf{W}(\mathbf{x} - \mathbf{\Phi}(\hat{\mathbf{a}})\hat{\mathbf{r}})\|_2^2, \tag{17}$$

where $\mathbf{W}$ is the matrix that performs the projection in the perceptual domain.

The problem is now how to define the rate. Since the definition of the distortion is related to the selection of $\hat{\mathbf{a}}$ and $\hat{\mathbf{r}}$, we can split the rate accordingly:

$$R(\hat{\mathbf{x}}) = R(\hat{\mathbf{a}}) + R(\hat{\mathbf{r}}). \tag{18}$$

We can then consider the cardinality of the two vectors as a coarse approximation of the rate, and rewrite the rate as:

$$R(\hat{\mathbf{x}}) = \alpha \|\hat{\mathbf{a}}\|_0 + \beta \|\hat{\mathbf{r}}\|_0. \tag{19}$$

Considering now (15) and the alternative LPAS formulation presented in Section 3, the operational rate distortion problem becomes:

$$\hat{\mathbf{a}}_k, \hat{\mathbf{r}}_k = \underset{\mathbf{a},\mathbf{r}}{\operatorname{argmin}} \|\mathbf{r} - \check{\mathbf{x}} - \check{\mathbf{X}}\mathbf{a}\|_2^2 + \alpha\|\mathbf{a}\|_0 + \beta\|\mathbf{r}\|_0, \tag{20}$$

where we use Lagrange multipliers to define the unconstrained minimization problem and we imposed $\mathbf{W} = \mathbf{I}$. We can now see the similarities to our new LPAS formulation in (13). The problem become *equivalent* when $p = 2$ and the cardinality (the 0-norm) is approximated with the 1-norm, as done in, e.g., [11].

4. REAL-TIME CONVEX FORMULATION

Several methods exist for solving (13) efficiently. In this case, given that the objective is smooth with $p = 2$ and the gradient and the projection onto the set $\{z \in \mathbb{R}^n \,|\, \|z\|_1 \leq \rho\}$ can be calculated efficiently, we employ gradient projection methods (a special case of proximal gradient methods). These methods have been used extensively in signal processing applications [20–25] and, more recently, for real-time signal processing [26, 27].

Firstly, we cast the problem (13) into the form

$$\begin{array}{ll} \text{minimize} & f(\mathbf{z}) \\ \text{subject to} & \mathbf{z} \in \mathbb{Z}\,. \end{array} \tag{21}$$

While several variants of fast proximal/projection gradient method exist for problems of the form (21), we choose [28, 2.2.19] for simplicity:

$$\mathbf{z}^{(j+1)} = P_{\mathbb{Z}}\left(\mathbf{y}^{(j)} - \frac{1}{L}\nabla f(\mathbf{y}^{(j)})\right) \tag{22}$$

$$\alpha_{j+1}^2 = \tfrac{1}{2}(-\alpha_j^2 + \sqrt{\alpha_j^4 + 4\alpha_j^2}) \tag{23}$$

$$\beta_j = \frac{\alpha_j(1-\alpha_j)}{\alpha_j^2 + \alpha_{j+1}} \tag{24}$$

$$\mathbf{y}^{(j+1)} = \mathbf{z}^{(j+1)} + \beta_j(\mathbf{z}^{(j+1)} - \mathbf{z}^{(j)}) \tag{25}$$

where $\mathbf{y}^{(j)}$ is an additional iteration vector of the same size as $\mathbf{z}^{(j)}$, $\alpha_j, \beta_j \in \mathbb{R}$ and the Euclidean projection $P_{\mathbb{X}}(\mathbf{z})$ of $\mathbf{z}$ onto the set $\mathbb{Z}$ is given by

$$P_{\mathbb{Z}}(\mathbf{z}) = \underset{\mathbf{y}\in\mathbb{Z}}{\operatorname{argmin}} \|\mathbf{z} - \mathbf{y}\|_2^2\,. \tag{26}$$

The complexity of this method is dominated by the projection $P_{\mathbb{Z}}(z)$ and calculation of the gradient $\nabla f(\mathbf{z})$. In the following we show how these steps can be computed with linearithmic time complexity $\mathcal{O}(N \log N)$.

Acknowledging that the feasible set in (13) is separable, $\mathbf{a} \in \mathbb{R}^N, \mathbf{r} \in \mathbb{R}^{N+1}$ can be projected in two independent steps. The projection onto the set $\{\mathbf{z} \in \mathbb{R}^n \,|\, \|\mathbf{z}\|_1 \leq \rho\}$ is given as

$$P_{\{\mathbf{z}\,|\,\|\mathbf{z}\|_1 \leq \rho\}}(\mathbf{z}) = S_\lambda(\mathbf{z}) \tag{27}$$

where S is the soft-thresholding function

$$(S_t(\mathbf{v}))_i = \operatorname{sgn}(\mathbf{v}_i)\max(|\mathbf{v}_i| - t, 0) \tag{28}$$

and $\lambda = 0$ if $\|\mathbf{z}\|_1 \leq \rho$, otherwise λ is the solution to the non-linear equation

$$\sum_{i=1}^{n} \max(|\mathbf{z}_i| - \lambda, 0) = \rho\,. \tag{29}$$

Algorithms for solving this non-linear equations often involves sorting $\mathbf{z} \in \mathbb{R}^n$ in magnitude followed by some arithmetic with no more than $\mathcal{O}(n)$ time complexity. Using an optimal comparison sorting method this approach then has the worst-case time complexity of $\mathcal{O}(N \log N)$ and has been suggested independently in several publications [29, 30], and, more recently, [31].

The gradient is given by

$$\begin{aligned} \nabla f(\mathbf{z}) &= \nabla f(\mathbf{a}_k, \mathbf{r}_k) \\ &= \begin{bmatrix} \check{\mathbf{X}}_k & -I \end{bmatrix}^T \left(\begin{bmatrix} \check{\mathbf{X}}_k & -I \end{bmatrix} \begin{bmatrix} \mathbf{a}_k \\ \mathbf{r}_k \end{bmatrix} + \check{\mathbf{x}}_k \right). \end{aligned} \tag{30}$$

The matrix multiplications, e.g., $\check{\mathbf{X}}_k\mathbf{a}$, can be also calculated using Fast Fourier Transform (FFT) filtering with the time complexity $\mathcal{O}(N \log N)$.

Let J be the number of iterations, then this algorithm requires $4J$ FFTs (4 per iteration for FFT filtering) and $2J$ sorting operations.

5. EXPERIMENTAL ANALYSIS

We evaluated the coding scheme in (13) in terms of coding efficiency and compared it with a traditional CELP scheme [32]. We remark that we are not here interested in defining the best way to encode predictor and residual but rather show that better ways to allocate bits between predictor and residual are possible.

We analyzed one hour of clean speech extracted from the TIMIT database. We chose speakers with different characteristics (gender, age, pitch, regional accent). Without loss of generality, we resampled at 8 kHz to compare our method with G.711 and the CELP formulation in [32]. The frame length was 160 samples (20 ms). The order of the LP filter is 159, according to Section 3. This means we can cover accurately pitch frequencies in the interval [70 Hz, 500 Hz].

The parameters δ and γ in (13) were chosen empirically based on 50% of the data. The rest of the data was used to perform the experimental analysis. The speech signal were normalized at -26 dBFS not to have level dependencies in the parameters. We used three different versions of our sparse LP, corresponding to different values of the parameters in the optimization criterion. Finding tradeoffs between the parameters is not easy, differently from traditional sparse linear prediction [11], there are not close form bounds for δ and γ. We chose empirically the set of values below.

METHOD	δ	γ
$\text{SpLPAS}_{\text{v1}}$	2	.30
$\text{SpLPAS}_{\text{v2}}$	1.9	.27
$\text{SpLPAS}_{\text{v3}}$	1.7	.21

In order to give an indication of the running time for the algorithm of Section 4, we implemented it in C++ using Math Kernel Library vector, BLAS level 1 functions, and FFTs using FFTW3 [33] running on an Intel(R) Core(TM)

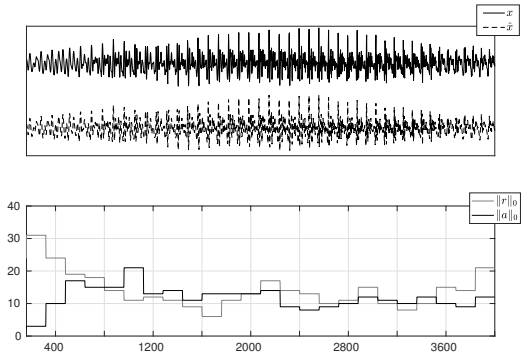

Fig. 1. The top pane shows the original and reconstructed segment of voiced speech. The bottom pane shows the cardinlaity of the estimated $\mathbf{r}$ and $\mathbf{a}$.

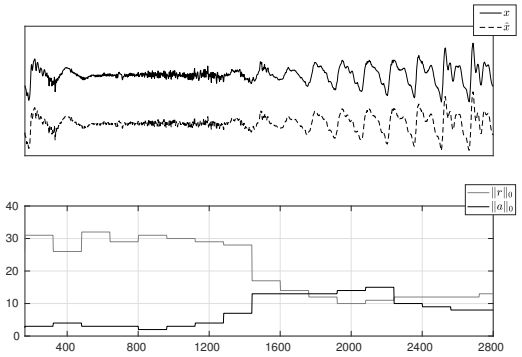

Fig. 2. The top pane shows the original and reconstructed segment of unvoiced speech. The bottom pane shows the cardinality of the estimated $\mathbf{r}$ and $\mathbf{a}$.

i5-2410M CPU at 2.30GHz[2]. With this implementation, it took approximately $60\,[\mu\mathrm{s}]$ to run an iteration or approximately $1.7\,[\mathrm{ms}]$ in total for $J = 30$ iterations and frame size $N = 160$ compared to $934.0\,[\mathrm{ms}]$ using an interior-point method (CVX+SeDuMi), two order of magnitudes faster.

In order to quantify the bit rate, we coded the sparse residual and sparse predictor using a mix of parametric and non-parametric modeling, as done in [34]. The average rate was 9 bits per residual coefficient and 14 bits per prediction coefficient. The locations of both coefficients were coded as a memoryless random process with $\log_2 \binom{N-1}{\|\mathbf{a}\|_0}$ and $\log_2 \binom{N}{\|\mathbf{r}\|_0}$ for predictor and residual, respectively. We used $32\times32\rightarrow64$ bit operations for the decoder. An example for voiced and unvoiced speech is shown in Figure 1 and Figure 2. It is particularly interesting how the cardinality changes in Figure 2, from a unvoiced segment to a more voiced part. In the beginning the $\mathbf{r}$ carries most of the information, hence the high cardinality, then drops to make more room for the predictor $\mathbf{a}$, consistently with the increase in sample correlation.

Finally, we compared our LPAS methods (SpLPAS) obtained with different set of parameters with the ITU G.711 waveform coder working at 64 kb/s (μ-law PCM) and the low-complexity CELP presented in [32]. It is easy to see that our coding scheme allows for generally better bit allocation between the two components (predictor and residual). Although, the bit rate obtained by the SpLPAS is currently not competitive, better coders can be built by also exploiting the characteristics of residual and predictor. For example, for voiced speech, the high-order predictor could be decomposed into short-term and long-term components and coded using traditional methods (e.g., line spectral frequencies [1]). Furthermore, we are not applying any perceptual weighting which would help reducing the bit rate dramatically.

[2]Code for the proposed algorithm SpLPAS available at `https://github.com/giacobello/SpLPAS`

Table 1. Comparison in terms of average cardinality ($\|\mathbf{a}\|_0$, $\|\mathbf{r}\|_0$), average bit rate for each component ($R_{\mathbf{a}}$, $R_{\mathbf{r}}$), total average bit rate R_{tot}, and Mean Opinion score for the speech coding algorithm considered.

METHOD	$\|\mathbf{a}\|_0$	$\|\mathbf{r}\|_0$	$R_{\mathbf{a}}$	$R_{\mathbf{r}}$	R_{tot}	MOS
G.711	N/A				64	4.22
CELP [32]	10+1	26	4	12	16	3.97
SpLPAS$_{\mathrm{v1}}$	15	17	18.1	15.3	38.8	4.07
SpLPAS$_{\mathrm{v2}}$	13	14	15.6	12.6	32.9	3.84
SpLPAS$_{\mathrm{v3}}$	10	11	12.3	10.1	23.2	3.29

6. CONCLUSIONS

We have introduced an alternative formulation for the analysis-by-synthesis linear prediction scheme commonly used in speech codecs. This formulation allows for a one-step estimation of predictor and residual with the possibility of choosing the right distortion level and sparsity, intimately related to the bit rate, for each speech frame. The idea to perform lossy coding only in the optimization step, where we have full control over the parametrization, and then proceed to encode the predictor and residual losslessly. This allows to choose better representations (and, in turn, bit allocations) between predictor and residual according to the nature of the speech segment.

7. REFERENCES

[1] P. Kroon and W. B. Kleijn, "Linear-prediction based analysis-by-synthesis coding," in *Speech coding and Synthesis*, pp. 79–119, 1995.

[2] J.-H. Chen and J. Thyssen, "Analysis-by-synthesis speech coding," in *Springer Handbook of Speech Processing*, pp. 351–392, 2008.

[3] K. K. Paliwal and B. S. Atal, "Efficient vector quantization of

LPC parameters at 24 bits/frame," *IEEE Trans. on Speech and Audio Processing*, vol. 1, no. 1, pp. 3–14, 1993.

[4] B. Bessette, R. Salami, R. Lefebvre, M. Jelinek, J. Rotola-Pukkila, J. Vainio, H. Mikkola, and K. Jarvinen, "The adaptive multirate wideband speech codec (AMR-WB)," *IEEE Trans. on Speech and Audio Processing*, vol. 10, no. 8, pp. 620–636, 2002.

[5] F.F. Tzeng, "Near-optimum linear predictive speech coding," in *Proc. GLOBECOM*, pp. 962–966, 1990.

[6] J. P. Woodard and L. Hanzo, "Improvements to the analysis-by-synthesis loop in CELP codecs," in *Proc. Intl. Conf. on Radio Receivers and Associated Systems*, pp. 114–118, 1995.

[7] S. Singhal and B.S. Atal, "Optimizing LPC filter parameters for multi-pulse excitation," in *Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing*, vol. 8, pp. 781–784, 1983.

[8] M. Fratti, G.A. Mian, and G. Riccardi, "On the effectiveness of parameter reoptimization in multipulse based coders," in *Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 73–76, 1992.

[9] D. Giacobello, M. N. Murthi, M. G. Christensen, S. H. Jensen, and M. Moonen, "Re-estimation of linear predictive parameters in sparse linear prediction," in *Rec. Asilomar Conf. on Signals, Systems and Computers*, pp. 1770–1773, 2009.

[10] D. Giacobello, T. van Waterschoot, M.G. Christensen, S. H. Jensen, and M. Moonen, "High-order sparse linear predictors for audio processing," in *Proc. European Signal Processing Conf.*, pp. 234–238, 2010.

[11] D. Giacobello, M. G. Christensen, M. N. Murthi, S. H. Jensen, and M. Moonen, "Sparse linear prediction and its applications to speech processing," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 20, no. 5, pp. 1644–1657, 2012.

[12] M. N. Murthi and B. D. Rao, "Towards a synergistic multistage speech coder," in *Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing*, pp. 369–372, 1998.

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

[14] W. B. Kleijn, D. Krasinski, and R. Ketchum, "Fast methods for the CELP speech coding algorithm," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 38, no. 8, pp. 1330–1342, 1990.

[15] T. Bäckström, "Comparison of windowing in speech and audio coding," in *Proc. Workshop on Applications of Signal Processing to Audio and Acoustics*, 2013.

[16] J. Fischer and T. Bäckström, "Comparison of windowing schemes for speech coding," in *Proc. European Signal Processing Conf.*, pp. 804–808, 2015.

[17] Y. Shoham, A. Gersho, A., "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 36, no. 9, pp.1445–1453, 1988.

[18] P. Prandoni and M. Vetterli, "R/D optimal linear prediction," *IEEE Trans. on Speech and Audio Processing*, vol. 8, no. 6, pp. 646–655, 2000.

[19] C. Weidmann and M. Vetterli, "Rate distortion behavior of sparse sources," *IEEE Trans. on Information Theory*, vol. 58, no. 8, pp. 4969–4992, Aug 2012.

[20] P. Combetttes and V. Wajs, "Signal recovery by proximal forward-backward splitting," *Multiscale Model. Simul.*, vol. 4, pp. 1168–1200, 2005.

[21] J. M. Bioucas-Dias and M. A. T. Figueiredo, "A new TwIST: Two-step iterative shrinkage/thresholding algorithms for image restoration," *IEEE Trans. Image Processing*, vol. 16, no. 12, pp. 2992–3004, 2007.

[22] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo, "Sparse reconstruction by separable approximation," *IEEE Transaction on Signal Processing*, vol. 57, pp. 2479–2493, 2009.

[23] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sci.*, vol. 2, pp. 183–202, 2009.

[24] J. Dahl, P. C. Hansen, S. H. Jensen, and T. L. Jensen, "Algorithms and software for total variation image reconstruction via first-order methods," *Numerical Algorithms*, vol. 53, pp. 67–92, 2010.

[25] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, 2010.

[26] B. Defraene, T. van Waterschoot, H. J. Ferreau, M. Diehl, and M. Moonen, "Real-time perception-based clipping of audio signals using convex optimization," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 20, no. 10, pp. 2657–2671, 2012.

[27] B. Defraene, T. van Waterschoot, M. Diehl, and M. Moonen, "Embedded-optimization-based loudspeaker precompensation using a Hammerstein loudspeaker model," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 22, no. 11, pp. 1648–1659, 2014.

[28] Y. Nesterov, *Introductory Lectures on Convex Optimization, A Basic Course*, Kluwer Academic Publishers, 2004.

[29] E. J. Candés and J. Romberg, "Practical signal recovery from random projections," in *Intl. Symposium on Electronic Imaging*, 2005.

[30] E. van den Berg and M. P. Friedlander, "Probing the Pareto frontier for basis pursuit solutions," *SIAM Journal on Scientific Computing*, vol. 31, no. 2, pp. 890–912, 2008.

[31] J. Songsiri, "Projection onto an l1-norm ball with application to identification of sparse autoregressive models," in *Proc. Asean Symposium on Automatic Control*, 2011.

[32] J.-H. Chen, "Toll-quality 16 kb/s CELP speech coding with very low complexity," in *Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 9–12, 1995.

[33] M. Frigo and S. G. Johnson, "The design and implementation of FFTW3," in *Proc. IEEE*, vol. 93, pp. 216–231, 2005.

[34] F. Ghido and I. Tabus, "Sparse modeling for lossless audio compression," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 21, no. 1, pp. 14–28, 2013.