Available online at www.sciencedirect.com

ScienceDirect

Speech Communication 76 (2016) 143–156

SPEECH COMMUNICATION

www.elsevier.com/locate/specom

Fast algorithms for high-order sparse linear prediction with applications to speech processing

Tobias Lindstrøm Jensen [a,*,1], Daniele Giacobello [a], Toon van Waterschoot [b], Mads Græsbøll Christensen [c]

[a] Signal and Information Processing, Department of Electronic Systems, Aalborg University, Denmark
[b] Center for Dynamical Systems Signal Processing and Data Analytics (STADIUS), Department of Electrical Engineering (ESAT), KU Leuven, Belgium
[c] Audio Analysis Lab, AD:MT, Aalborg University, Denmark

Received 9 January 2015; received in revised form 7 September 2015; accepted 21 September 2015
Available online 19 October 2015

Abstract

In speech processing applications, imposing sparsity constraints on high-order linear prediction coefficients and prediction residuals has proven successful in overcoming some of the limitation of conventional linear predictive modeling. However, this modeling scheme, named sparse linear prediction, is generally formulated as a linear programming problem that comes at the expenses of a much higher computational burden compared to the conventional approach. In this paper, we propose to solve the optimization problem by combining splitting methods with two approaches: the Douglas–Rachford method and the alternating direction method of multipliers. These methods allow to obtain solutions with a higher computational efficiency, orders of magnitude faster than with general purpose software based on interior-point methods. Furthermore, computational savings are achieved by solving the sparse linear prediction problem with lower accuracy than in previous work. In the experimental analysis, we clearly show that a solution with lower accuracy can achieve approximately the same performance as a high accuracy solution both objectively, in terms of prediction gain, as well as with perceptually relevant measures, when evaluated in a speech reconstruction application.

Keywords: Sparse linear prediction; Speech and audio processing; Linear programming; Real-time optimization; Speech reconstruction; Packet loss concealment

1. Introduction

Linear prediction (LP) is a well understood technique for the analysis, modeling, and coding of speech signals (Vaidyanathan, 2009). The widespread use of LP of speech can be attributed to its correspondence to the source-filter model of speech production (Makhoul, 1975; Bäckström, 2004). An emitted speech sound can be modeled as a combination of the excitation process (the air flow) and the filtering process (vocal tract effect). The vocal tract can, to a large extent, be modeled as a slow varying low-order all-pole filter, while the air flow can be modeled by a white noise sequence, for unvoiced sounds, or an impulse train generated by periodic vibrations of the vocal chords pulses, for voiced sounds (Hansen et al., 1987).

In speech analysis, the purpose of the all-pole model obtained through LP is to construct a spectral envelope that models the behavior of the vocal tract. For a segment of unvoiced speech, considering the excitation of the

* Corresponding author.
E-mail addresses: tlj@es.aau.dk (T.L. Jensen), giacobello@ieee.org (D. Giacobello), toon.vanwaterschoot@esat.kuleuven.be (T. van Waterschoot), mgc@create.aau.dk (M.G. Christensen).
[1] The work of T.L. Jensen is supported by the Danish Council for Independent Research, Technology and Production Sciences. Grant No. 4005-00122.

http://dx.doi.org/10.1016/j.specom.2015.09.013
0167-6393/

all-pole filter as white noise, the envelope is the same as its power spectrum of and the LP model coincides theoretically with the autoregressive (AR) model (Stoica and Moses, 2005). However, for a segment of voiced speech, the connection is more complex. The power spectrum of the voiced speech signal has a clear harmonic structure that can be approximated more effectively as a line spectrum (Christensen and Jakobsson, 2009). The line frequencies are located at the multiples of the pitch frequency and their amplitude are given by the shape of the spectral envelope.

The all-pole coefficients are usually identified by minimizing the mean-squared (2-norm) error of the difference between the observed signal and the predicted signal (Atal and Hanauer, 1971). In the source-filter model, this approach yields the LP all-pole filter, thus the prediction error (the residual signal) represents the source. Unvoiced speech lends itself readily to the principles of the 2-norm error criterion as a means of estimating the model parameters (Makhoul, 1975). Furthermore, the 2-norm is consistent with an i.i.d. Gaussian interpretation of the prediction residual (Saito and Itakura, 1967; Itakura and Saito, 1970). The quality of the 2-norm based LP all-pole model in the context of voiced speech, which is approximately two-thirds of speech, is questionable and, theoretically, not well-founded. In particular, the all-pole spectrum does not provide a good spectral envelope and sampling the spectrum at the line frequencies does not provide a good approximation of their amplitudes (Murthi and Rao, 2000). In general, the shortcomings of LP in spectral envelope modeling can be traced back to the 2-norm minimization.[2] In particular, analyzing the goodness of fit between a given harmonic line spectrum and its LP model, as done in Makhoul (1975), a major flaw can be derived. The LP tries to cancel the input voiced speech harmonics causing the resultant all-pole model to have poles close to the unit circle. Consequently, the LP spectrum tends to overestimate the spectral powers at the formants, providing a sharper contour than the original vocal tract response. A wealth of methods have been proposed to mitigate these effects. Some of the proposed techniques involve a general rethinking of the spectral modeling problem (notably El-Jaroudi and Makhoul, 1991; Murthi and Rao, 2000; Ekman et al., 2008) while some others are based on changing the statistical assumptions made on the prediction error in the minimization process (notably Lee, 1988; Denoel and Solvay, 1985). Many other formulations for finding the parameter of the all-pole model exist, a special mention is for methods that include perceptual knowledge into the estimation process (e.g., Hermansky, 1990; Magi et al., 2009), or account for the non-linearities in the speech production model, e.g., Thyssen et al. (1994).

Despite the wealth of alternative methods introduced to overcome the deficiencies of the 2-norm criterion, traditional usage of LP methods is, however, still confined to modeling only the spectral envelope (the vocal tract transfer function), i.e., the short-term redundancies of speech. Hence, in the case of voiced speech, the predictor does not fully decorrelate the speech signal because of the long-term redundancies of the underlying pitch excitation. This means that the residual will still have pitch pulses present and the spectrum will still show a clear harmonic structure. The usual approach is then to employ a cascaded structure where, after LP is initially applied to determine the short-term prediction coefficients, a long-term predictor is determined to model the harmonic behavior of the spectrum (Hansen et al., 1987). Such a structure is arguably suboptimal since it ignores the interaction between the two different stages (Kameoka et al., 2010; Bensaid and Slock, 2012). This is known in the literature and early contributions have outlined gains in performance in jointly estimating the two filters (the work in Kabal and Ramachandran (1989) is perhaps the most successful attempt). The combination of the two filters determines a high-order linear predictor with a pretty evident sparse characteristics.

In recent work (Giacobello et al., 2008; Giacobello et al., 2012), a more general framework for LP was presented with several benefit by introducing sparsity in the LP minimization framework. This was renamed sparse linear prediction (SpLP). In particular, while reintroducing well-known methods to seek a short-term predictor that produces a residual that is sparse rather than minimum variance (e.g., Denoel and Solvay, 1985; Murthi and Rao, 1998), the idea of employing high-order SpLP (HOSpLP) to model the cascade of short-term and long-term predictors was also introduced (Giacobello et al., 2009,). The application of HOSpLP was originally introduced for speech processing purposes, however its formulation is intimately related to the regularization of ill-conditioned problems and to the precise modeling of long-term redundancies, thus it quickly found applications in diverse fields, such as radar (Erer et al., 2014), geology (Bochud et al., 2013), video packet-loss concealment (Koloda et al., 2013), and general signal representations (Angelosante et al., 2013; Angelosante, 2014).

The SpLP problem can be posed as a linear programming problem, a special case of convex optimization. In order to be deployed in real-time applications, it requires its convex optimization core to be embedded directly in the algorithm that runs online and where strict real-time constraints apply. While convex optimization problems can be efficiently solved, both in theory, with worst-case polynomial complexity (Nesterov and Nemirovskii, 1994), and in practice, such as Andersen et al. (2003), it is rarely

[2] To the authors' knowledge, the "original sin" behind the use of the 2-norm in LP, comes from its first application in speech coding, trying to reduce the entropy of speech for more efficient encoding than simple differential pulse code modulation (Atal, 2006). The fundamental theorem of predictive quantization (Gersho and Gray, 1992) states that the mean-squared reproduction error in predictive encoding is equal to the mean-squared quantization error when the residual signal is presented to the quantizer. Therefore, by minimizing the 2-norm of the residual, these variables have a minimal variance whereby the most efficient coding is achieved.

limited in its implementation by real-time constraints as discussed in Mattingley and Boyd (2010). The real-time implementation of such schemes calls for application-tailored optimization methods able to solve instances of the optimization problem at hand in a fast and reliable way (see, e.g., Defraene et al., 2012; Jensen et al., 2013; Defraene et al., 2014 for application in signal processing). Convex optimization solvers are usually based on iterative approaches, which is in contrast to traditional methods relaying on closed-form solutions. This is also the case for LP and SpLP. The former, has a closed-form solution that, e.g., can be calculated via the Levinson–Durbin algorithm with time complexity $\mathcal{O}(N^2)$, with N being the prediction order. The latter, when solved with, e.g., interior-point methods (Alipoor and Savoji, 2012; Jensen et al., 2013), has a time complexity of $\mathcal{O}(T^2N+T^3)$ or $\mathcal{O}(N^2T+N^3)$ depending on the setup where T is the frame size. Thus, considering the high-order case where $N \approx T$, the sparse solution is at least a order of magnitude more costly to achieve.

To reduce the complexity of solving the SpLP problem, we turn our attention to two other methods, specifically the Douglas–Rachford (DR) method and the alternating direction method of multipliers (ADMM). These two methods applied with a splitting technique that have become popular in recent years for problems requiring only moderate accuracy, see e.g. the treatment in Boyd et al. (2011) and Vandenberghe (2013). The DR method originates from Douglas and Rachford (1956) and Lions and Mercier (1979), and have recently found applications in signal processing problems, e.g., Combettes and Pesquet (2007) and O'Connor and Vandenberghe (2014). Similar, the ADMM method originates from Glowinski and Marroco (1975) and Gabay and Mercier (1976) and have also found applications in signal processing, e.g., Afonso et al. (2010) and Yang and Zhang (2011). Interestingly, there are known connections among proximal methods, Bregman iterative regularization, and the DR and ADMM algorithms. Specifically, ADMM can be understood as the DR method applied to the dual problem (Gabay, 1983; Eckstein and Bertsekas, 1992; Yin et al., 2008).

In this paper, we will show how to reduce the per-iteration time complexity for an iterative solver for the SpLP problem employing splitting methods rather than interior-point methods. The splitting methods require solving a subproblem involving a symmetric positive definite Toeplitz coefficient matrix. By exploiting this particular structure, the time complexity is quadratic ($\mathcal{O}(N^2+T^2)$) for the initialization step but linearithmic ($\mathcal{O}(N\log N + T\log T)$) for all the subsequent iterations. In order to evaluate the approximate solutions, firstly, we assess their performance via prediction gain and, secondly, assess the performance in terms of perceptual objective quality measures in a speech reconstruction framework. Despite solving the SpLP problem to a lower accuracy compared to interior-point methods, the results show that the solutions still achieve similar performance when employed in typical speech processing applications.

The paper is organized as follows. In Section 2, we present principles of linear prediction including conventional methods and sparse linear prediction. The proposed methods for solving the sparse linear prediction problem are presented in Section 3, and the computational costs are assessed by timings the methods in Section 4. Finally, we present experimental results of the performance of the presented predictors both in objective terms by analyzing their prediction gains in Section 5.1 and in perceptual terms by employing them to reconstruct missing data in a speech reconstruction framework in Section 5.2. In Section 6, we provide additional discussions and conclude our work.

2. Principles of linear prediction

Linear prediction is based on the following model, where a stationary set of samples of speech $x[t]$, for $t = 1, \ldots, T$, are written as a linear combination of N past samples (Vaidyanathan, 2009)

$$x[t] = \sum_{n=1}^{N} \alpha_n x[t-n] + r[t], \tag{1}$$

where $\{\alpha_n\}$ are the prediction coefficients and $r[t]$ is the prediction error. The optimization problem is then to find the prediction coefficient vector $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_N]^T$ so that the prediction error in this interval is minimized (Stoica and Moses, 2005). If we rewrite this model for the segment of T samples considered in matrix form

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

the optimization problem becomes

$$\underset{\alpha}{\text{minimize}} \quad \|x - X\alpha\|_p^p \tag{3}$$

where $\|\cdot\|_p$ is the p-norm defined as $\|\mathbf{x}\|_p = (\sum_{n=1}^{N}|x(n)|^p)^{\frac{1}{p}}$ for $p \geqslant 1$ and

$$[x|X] = \begin{bmatrix} x[T_1] & \cdots & x[T_1-n] \\ \vdots & \ddots & \vdots \\ x[T_2] & \cdots & x[T_2-n] \end{bmatrix}. \tag{4}$$

Assuming $x[t] = 0$ for $t < 1$ and $t > T$, the indexes T_1 and T_2 can be chosen in various ways which lead to different types of solutions with different properties (Stoica and Moses, 2005).

2.1. Conventional linear prediction

In a common case, the starting and ending point of the window used to determine the set of equation in (3) are chosen as $T_1 = 1$ and $T_2 = T + N$ and the 2-norm is minimized ($p = 2$). This leads to the conventional linear prediction problem

$$\underset{\alpha}{\text{minimize}} \quad \|x - X\alpha\|_2^2 \tag{5}$$

with $x \in \mathbb{R}^{M\times 1}$, $X \in \mathbb{R}^{M\times N}$, $\alpha \in \mathbb{R}^{N\times 1}$, $M = T + N$, and an analytic solution satisfying the normal equation

$$X^T X\alpha = X^T x. \tag{6}$$

This approach is also known as the Yule–Walker method for autoregressive (AR) spectral estimation or the autocorrelation method (Stoica and Moses, 2005). By exploiting the Toeplitz structure of the autocorrelation matrix $R = X^T X$, the system can be solved efficiently with, e.g., the Levinson–Durbin algorithm in $\mathcal{O}(N^2)$ (Vaidyanathan, 2009).

2.2. Long-term prediction

Generally, linear prediction models only short-term redundancies of speech, thus is often used in combination with a single-tap or multi-tap long-term predictor (Kabal and Ramachandran, 1989). The speech model for the long-term predictor is

$$d[t] = \sum_{k=0}^{K} \phi_k d[t - T_p - k] + r[t], \tag{7}$$

where, similarly to (1), $\{\phi_k\}$ are the prediction coefficients and $r[t]$ is the prediction error. The major difference of the model, other than the relatively small number of taps K employed gathered around the pitch period T_p is that the optimization problem is generally done on the (usually weighted) output of the short-term filter $d[t]$. Considering the matrix form of (7)

$$d = D\phi + r, \tag{8}$$

we can obtain the coefficients by solving the optimization problem

$$\underset{\phi}{\text{minimize}} \quad \|d - D\phi\|_2^2. \tag{9}$$

The method obviously requires an estimate of T_p which can be found with a variety of methods all with different complexity and accuracy trade-offs (Christensen and Jakobsson, 2009).

2.3. Combining short-term and long-term prediction

If we consider the cascade of the long-term and short-term predictor, it is not hard to see that a sparse high-order filter is obtained. This was noted already in Kabal and Ramachandran (1989), and used to find short-term and long-term predictors jointly. In particular, the sparse filter will have two well distinguished nonzero regions: the first N taps will correspond to the short-term predictor α while the taps after the pitch period T_p will correspond to the convolution between the short-term and long-term coefficients $\varrho = \alpha * \phi$.

This gives us the opportunity of seeing the estimation problem as

$$\underset{\upsilon}{\text{minimize}} \quad \|x - U\upsilon\|_2^2 \tag{10}$$

where U is a column-wise partition of X accounting only for the nonzero elements of the combined filter $\upsilon = [\alpha^T \ \varrho^T]^T$. Again, applying the same traditional linear prediction approach shown in (5), we obtain the normal equations

$$U^T U\upsilon = U^T x, \tag{11}$$

where $U^T U$ retains a Toeplitz structure and its size is much smaller than $X^T X$, as only a fraction of the elements in x is used to estimate the combined predictor υ.

2.4. High-order prediction

As mention earlier, LP of speech is well known mostly for its modeling capabilities of short-term redundancies, which corresponds to a model of the envelope of the spectrum. When the LP order increases, the model starts encompassing more and more details of the spectrum, thus allowing for a more complete frequency representation. This follows directly from the theory, where for $N \to \infty$ the spectrum of the predictor matches the one of the signal (Makhoul, 1975). This means that also in the voiced speech case, we can model the signal using just a high-order LP model without worrying about, e.g., pitch estimation. However, calculating a high-order predictor generally results in a ill-conditioned problem as the model order N approaches the length of the frame T (van Waterschoot and Moonen, 2008). The ill-conditioning can be easily tracked back to the autocorrelation matrix $R = X^T X$. In particular, the eigenvalue spread of the autocorrelation matrix corresponds to the ratio between maximum and minimum value of the power spectrum of the speech segment analyzed, therefore, except for few exceptions, a high order in LP for speech makes R easily ill-conditioned and can cause large variance of the estimated model parameters, leading to spurious peaks in the signal spectral estimate (Kay, 1979). While regularization is possible to reduce the eigenvalue spread and thus conditioning, this also corresponds to adding a noise floor, affecting the accuracy of the solution.

2.5. High-order sparse linear prediction

In our recent work (Giacobello et al., 2012), we generalized the optimization problem in (3) by adding a regularization criterion on the solution vector

$$\underset{\alpha}{\text{minimize}} \quad \|x - X\alpha\|_p^p + \gamma\|\alpha\|_l^l. \tag{12}$$

Considering the similarities with conventional high-order LP, imposing sparsity on the high-order predictor while retaining a 2-norm criterion on the prediction error, the problem can be seen as a more educated regularization approach that accounts for the sparsity of the predictor

resulting from modeling short-term and long-term redundancies (Giacobello et al., 2009)

$$\underset{\alpha}{\text{minimize}} \quad \|x - X\alpha\|_2^2 + \gamma\|\alpha\|_1. \tag{13}$$

However, when imposing sparsity on both the residual vector and high-order predictor, gains can been obtained both in terms of modeling and coding performance (Giacobello et al., 2009)

$$\underset{\alpha}{\text{minimize}} \quad \|x - X\alpha\|_1 + \gamma\|\alpha\|_1. \tag{14}$$

For problem (14) we denote a solution $\alpha^\star$, objective $f(\alpha) = \|x - X\alpha\|_1 + \gamma\|\alpha\|_1$ and optimal objective $f^\star = f(\alpha^\star)$. The regularization term γ in (14) can be seen as related to the prior knowledge of the prediction coefficients vector α. While sparsity is often measured by the cardinality $\mathbf{card}(x) = |\{i|x_i \neq 0\}|$, we use the more computational tractable 1-norm $\|\cdot\|_1$, which is known throughout the sparse recovery literature (see, e.g., Donoho and Elad, 2002) to perform well as a relaxation of the cardinality measure with equivalence in certain cases.

3. Solving the sparse linear prediction problem

The objective function in (14), as well as the terms that compose it, is convex but not differentiable, thus proximal gradients method are not directly applicable (Nesterov, 2007; Beck and Teboulle, 2009; Wright et al., 2009). The optimization can be solved as a general linear programming problem using interior-point methods (Alipoor and Savoji, 2012; Jensen et al., 2013). However, solving (14) using interior-point methods introduces certain diagonal matrices D_1, D_2 into the problem, such that when using the augmented form approach, it is required to solve a linear system of equations with the coefficient matrix $C = X^T D_1 X + D_2$ where D_1, D_2 change at each iteration. This makes it difficult to exploit the structure in X and $X^T X$ using direct method (see, e.g., O'Connor and Vandenberghe, 2014). In Jensen et al. (2013), the coefficient matrix C is explicit formed and solved via a Cholesky factorization followed by triangular solves with a per iteration complexity of $\mathcal{O}(M^2N + M^3)$ or $\mathcal{O}(N^2M + N^3)$ depending on the setup (Jensen et al., 2013).

In the following, it is showed how to exploit the DR and the ADMM methods to reduce the per-iteration complexity. Specifically, with the use of splitting combined with ADMM and DR methods, an auxiliary symmetric positive definite Toeplitz system arises that can be solved with a total per-iteration complexity of $\mathcal{O}(N \log N + M \log M)$ (forming right-hand side and solving the system). The algorithm for solving the system is described in the subsequent Section 3.3.

The saving in per-iteration complexity of the DR and ADMM methods reflects into a slower convergence than interior-point methods. However, when tailored to speech processing applications, a low-accuracy solution of the problem (14) obtained via the DR or ADMM method is sufficient for practical purposes, as we will show in Section 5.

3.1. Douglas–Rachford

In the following we will rewrite the SpLP into a form applicable to the DR algorithm. We will write the problem in (14) as

$$\underset{\alpha}{\text{minimize}} \quad f_1(\alpha) + f_2(X\alpha) \tag{15}$$

where $f_1(u) = \gamma\|u\|_1$ and $f_2(u) = \|x - u\|_1$. Introducing the variable splitting $h(u_1, u_2) = f_1(u_1) + f_2(u_2)$ the problem can be reformulated as the optimization problem

$$\begin{aligned} \underset{u_1,u_2}{\text{minimize}} \quad & h(u_1, u_2) \\ \text{subject to} \quad & u_2 = Xu_1. \end{aligned} \tag{16}$$

Before we proceed, we define two functions. The proximal mapping of a convex function f is given by (see, e.g., Moreau (1965) or Combettes and Wajs (2005) for a more recent treatment)

$$\mathbf{prox}_f(x) = \underset{u}{\text{argmin}} \left(f(u) + \frac{1}{2}\|u - x\|_2^2 \right). \tag{17}$$

The Euclidean projection of x onto a set $\mathbb{C}$ is $\mathcal{P}_\mathbb{C}(x)$ given by

$$\mathcal{P}_\mathbb{C}(x) = \underset{y \in \mathbb{C}}{\text{argmin}} \|x - y\|_2^2. \tag{18}$$

Using the indicator function $\mathcal{I}_\mathbb{C}$ of the set $\mathbb{C}$, we obtain

$$\mathcal{P}_\mathbb{C}(x) = \mathbf{prox}_{\mathcal{I}_\mathbb{C}}(x). \tag{19}$$

Let $\mathbb{Q} = \{[u_1, u_2]^T | u_2 = Xu_1\}$, the Douglas–Rachford splitting method applied to problems of the form

$$\begin{aligned} \underset{u \in \mathbb{R}^U}{\text{minimize}} \quad & h(u) \\ \text{subject to} \quad & u \in \mathbb{Q} \end{aligned} \tag{20}$$

can be written in a number of equivalent forms, including (O'Connor and Vandenberghe, 2014; Eckstein and Bertsekas, 1992; Spingarn, 1985)

$$u^{(k+1)} = \mathbf{prox}_{th}(z^{(k)}) \tag{21}$$

$$y^{(k+1)} = \mathcal{P}_\mathbb{Q}(2u^{(k+1)} - z^{(k)}) \tag{22}$$

$$z^{(k+1)} = z^{(k)} + \eta(y^{(k+1)} - u^{(k+1)}) \tag{23}$$

with the iterates $u^{(k)}$, $z^{(k)}$, $y^{(k)} \in \mathbb{R}^{U \times 1}$. Here $\eta \in \mathbb{R}$ is a relaxation parameter $0 < \eta < 2$, $t \in \mathbb{R}$, $t > 0$ is a step-size parameter, or, equivalently a scaling of the problem.

The individual steps for solving the sparse linear prediction problem using Douglas–Rachford splitting are described in the following. For the update in (21), notice that $\mathbf{prox}_{th}(u_1, u_2) = [\mathbf{prox}_{tf_1}(u_1), \mathbf{prox}_{tf_2}(u_2)]^T$, i.e., it is separable since the binding is done in the constraints. A classical result is that with $f_1(u_1) = \gamma\|u_1\|_1$

$$\mathbf{prox}_{tf_1}(v) = \mathcal{S}_{t\gamma}(v) \tag{24}$$

where S is the soft-thresholding function given by

$$(\mathcal{S}_t(v))_i = \text{sign}(v_i)\max(|v_i| - t, 0) \tag{25}$$

and $(\cdot)_i$ denotes the ith element. For $f_2(u) = \|x-u\|_1$ we have

$$\begin{aligned}\mathbf{prox}_{tf_2}(v) &= \underset{u}{\text{argmin}} \left(t\|x-u\|_1 + \frac{1}{2}\|u-v\|_2^2 \right) \\ &= x - \underset{w}{\text{argmin}} \left(t\|w\|_1 + \frac{1}{2}\|w-(x-v)\|_2^2 \right) \\ &= x - \mathbf{prox}_{t\|\cdot\|_1}(x-v) = x - \mathcal{S}_t(x-v). \end{aligned} \tag{26}$$

So, $\mathbf{prox}_{th}(u_1,u_2) = [\mathbf{prox}_{tf_1}(u_1), \mathbf{prox}_{tf_2}(u_2)]^T = [\mathcal{S}_t(u_1), x - \mathcal{S}_t(x-u_2)]^T$ can be calculated with complexity $\mathcal{O}(M+N)$. The projection $\mathcal{P}_{\mathbb{Q}}(v)$ is given by

$$\mathcal{P}_{\mathbb{Q}}(v) = \underset{u\in\mathbb{Q}}{\text{argmin}} \|u-v\|_2^2 \tag{27}$$

$$= \underset{u_2 = Xu_1}{\text{argmin}} \|u_1 - v_1\|_2^2 + \|u_2 - v_2\|_2^2. \tag{28}$$

This is a convex quadratic problem with the KKT conditions

$$u_1 - v_1 - X^T v = 0 \tag{29}$$

$$u_2 - v_2 + v = 0 \tag{30}$$

$$u_2 = Xu_1 \tag{31}$$

from which we obtain

$$0 = u_1 - v_1 - X^T v \tag{32}$$

$$= u_1 - v_1 - X^T(v_2 - u_2) \tag{33}$$

$$= u_1 - v_1 - X^T v_2 + X^T X u_1. \tag{34}$$

Hence we obtain the linear systems $(I + X^T X)u_1 = v_1 + X^T v_2$ and $u_2 = Xu_1$. The projection is then

$$\mathcal{P}_{\mathbb{Q}}(v) = \begin{bmatrix} I \\ X \end{bmatrix} (I + X^T X)^{-1}(v_1 + X^T v_2). \tag{35}$$

To compute (35) we need to solve a linear system of equations with coefficient matrix $I + X^T X$ and varying right-hand sides $(v_1 + X^T v_2)$. The coefficient matrix is positive definite, symmetric and Toeplitz. Specifically, let

$$R = X^T X = \begin{bmatrix} r_0 & r_1 & \cdots & r_{N-1} \\ r_1 & r_0 & \cdots & r_{N-2} \\ \vdots & & \ddots & \vdots \\ r_{N-1} & r_{N-2} & \cdots & r_0 \end{bmatrix}. \tag{36}$$

and

$$t_0 = 1 + r_0 \tag{37}$$

$$t_k = r_k, \quad k = 1, \ldots, N-1 \tag{38}$$

then the positive definite, symmetric Toeplitz matrix is

$$I + X^T X = \begin{bmatrix} t_0 & t_1 & \cdots & t_{N-1} \\ t_1 & t_0 & \cdots & t_{N-2} \\ \vdots & & \ddots & \vdots \\ t_{N-1} & t_{N-2} & \cdots & t_0 \end{bmatrix}. \tag{39}$$

It is well known that linear systems with a coefficient matrix given as (39) can be solved efficiently. We will discuss different methods in Section 3.3.

3.2. Alternating direction method of multipliers

A first step in deriving an ADMM algorithm consists in reformulating the problem in (14) as a basis pursuit problem, following the procedure in Yang and Zhang (2011). To this end, we first rewrite the unconstrained problem in (14) as an equality constrained problem

$$\begin{aligned} &\underset{\alpha,e}{\text{minimize}} && \|e\|_1 + \gamma\|\alpha\|_1 \\ &\text{subject to} && X\alpha + e = x \end{aligned} \tag{40}$$

where $e = x - X\alpha$ represents the linear prediction residual. Next, we perform a change of variables by stacking a scaled version of the linear prediction coefficient vector and the linear prediction residual in a new parameter vector

$$\tilde{z} = \begin{bmatrix} \gamma\alpha \\ e \end{bmatrix}. \tag{41}$$

This allows to reformulate the problem in (40) using $\gamma\|\alpha\|_1 = \|\gamma\alpha\|_1$ in terms of a single parameter vector as follows (Yang and Zhang, 2011)

$$\begin{aligned} &\underset{\tilde{z}}{\text{minimize}} && \|\tilde{z}\|_1 \\ &\text{subject to} && \tilde{X}\tilde{z} = \tilde{x} \end{aligned} \tag{42}$$

where

$$\tilde{X} = [X \quad \gamma I] \tag{43}$$

$$\tilde{x} = \gamma x. \tag{44}$$

In a second step, we write the basis pursuit problem in (42) in the ADMM form as explained in Boyd et al. (2011). To this end, we define the set $\mathbb{U} = \{\tilde{z} \in \mathbb{R}^{m+n} | \tilde{X}\tilde{z} = \tilde{x}\}$ and introduce an variable $\tilde{y} \in \mathbb{R}^{m+n}$ such that the basis pursuit problem can be split over $\tilde{z}$ and $\tilde{y}$,

$$\begin{aligned} &\underset{\tilde{z},\tilde{y}}{\text{minimize}} && \mathcal{I}_{\mathbb{U}}(\tilde{z}) + \|\tilde{y}\|_1 \\ &\text{subject to} && \tilde{z} - \tilde{y} = 0 \end{aligned} . \tag{45}$$

This problem formulation readily brings us to an ADMM algorithm defined by the iterations (Boyd et al., 2011)

$$\tilde{z}^{(k+1)} = \mathcal{P}_{\mathbb{U}}(\tilde{y}^{(k)} - \tilde{u}^{(k)}) \tag{46}$$

$$\tilde{y}^{(k+1)} = \mathcal{S}_{1/\rho}(\tilde{z}^{(k+1)} + \tilde{u}^{(k)}) \tag{47}$$

$$\tilde{u}^{(k+1)} = \tilde{u}^{(k)} + \tilde{z}^{(k+1)} - \tilde{y}^{(k+1)}. \tag{48}$$

Variables $\tilde{z}^{(k)}$ and $\tilde{y}^{(k)}$ denote iterates of the primal variables, $\tilde{u}^{(k)}$ denotes the scaled dual variable, and $\rho > 0$

is the augmented Lagrangian parameter. Similarities can then be seen with the DR algorithm in (21)–(23).

We will now focus on the $\tilde{z}$-update in (46), which is the most involved step of the algorithm. The $\tilde{z}$-update consists in the projection of the point $\tilde{y}^{(k)} - \tilde{u}^{(k)}$ onto the convex set $\mathbb{U}$ with the following KKT conditions

$$\tilde{z} + (\tilde{y}^{(k)} - \tilde{u}^{(k)}) - \tilde{X}^T\lambda = 0 \tag{49}$$

$$\tilde{X}\tilde{z} = \tilde{x}. \tag{50}$$

It is instructive to rewrite these KKT conditions in terms of the original parameter vectors α and e by substituting the variable definitions (41), (43), and (44) in the KKT system (49) and (50)

$$\gamma\alpha - (\tilde{y}_1^{(k)} - \tilde{u}_1^{(k)}) - X^T\lambda = 0 \tag{51}$$

$$e - (\tilde{y}_2^{(k)} - \tilde{u}_2^{(k)}) - \gamma\lambda = 0 \tag{52}$$

$$X\alpha + e = x \tag{53}$$

where

$$\tilde{y}^{(k)} = \begin{bmatrix} \tilde{y}_1^{(k)} \\ \tilde{y}_2^{(k)} \end{bmatrix}, \ \tilde{u}^{(k)} = \begin{bmatrix} \tilde{u}_1^{(k)} \\ \tilde{u}_2^{(k)} \end{bmatrix}, \tag{54}$$

have been partitioned similarly to $\tilde{z}$ in (41). From the KKT conditions we obtain

$$0 = \gamma\alpha - (\tilde{y}_1^{(k)} - \tilde{u}_1^{(k)}) - X^T\lambda \tag{55}$$

$$= \gamma\alpha - (\tilde{y}_1^{(k)} - \tilde{u}_1^{(k)}) - \frac{1}{\gamma}X^T(e - (\tilde{y}_2^{(k)} - \tilde{u}_2^{(k)})) \tag{56}$$

$$= \gamma\alpha - (\tilde{y}_1^{(k)} - \tilde{u}_1^{(k)}) - \frac{1}{\gamma}X^T(x - X\alpha - (\tilde{y}_2^{(k)} - \tilde{u}_2^{(k)})). \tag{57}$$

This results in the following system to be solved for the linear prediction coefficient vector α

$$\begin{aligned}(X^TX + \gamma^2 I)\alpha &= X^T + \gamma(\tilde{y}_1^{(k)} - \tilde{u}_1^{(k)}) - X^T(\tilde{y}_2^{(k)} - \tilde{u}_2^{(k)}) \\ &= X^Tx - \begin{bmatrix} -\gamma I & X^T \end{bmatrix}(\tilde{y}^{(k)} - \tilde{u}^{(k)}).\end{aligned} \tag{58}$$

The solution $\alpha^{(k+1)}$ to this system in the $(k+1)$th ADMM iteration can be expressed as the sum of an iteration-independent term and an iteration-dependent term,

$$\alpha^{(k+1)} = \underbrace{(X^TX + \gamma^2 I)^{-1}X^Tx}_{\triangleq \alpha_{\gamma,2}} - \underbrace{(X^TX + \gamma^2 I)^{-1}\begin{bmatrix} -\gamma I & X^T \end{bmatrix}}_{\triangleq \begin{bmatrix} -\gamma I \\ X \end{bmatrix}^+}(\tilde{y}^{(k)} - \tilde{u}^{(k)}) \tag{59}$$

where $(\cdot)^+$ denotes the Moore–Penrose pseudo-inverse. The iteration-independent term $\alpha_{\gamma,2}$ is the solution to the ℓ_2-regularized linear prediction problem in (12) with $p = l = 2$. This system may, e.g., be solved by applying the Levinson–Durbin algorithm to the modified autocorrelation sequence $\{r_0 + \gamma^2, r_1, \ldots, r_N\}$ with r_k as defined in (36) for $k = 0, \ldots, N-1$ and similarly for $k = N$. The iteration-dependent term in (59) can be computed by solving a positive definite, symmetric Toeplitz system similar to the system solved in the DR method.

The ADMM iterations (46) and (48) can hence be rewritten as follows

$$\alpha^{(k+1)} = \alpha_{\gamma,2} - \begin{bmatrix} -\gamma I \\ X \end{bmatrix}^+(\tilde{y}^{(k)} - \tilde{u}^{(k)}) \tag{60}$$

$$e^{(k+1)} = x - X\alpha^{(k+1)} \tag{61}$$

$$\tilde{y}^{(k+1)} = \mathcal{S}_{1/\rho}\left(\begin{bmatrix} \gamma\alpha^{(k+1)} \\ e^{(k+1)} \end{bmatrix} + \tilde{u}^{(k)}\right) \tag{62}$$

$$\tilde{u}^{(k+1)} = \tilde{u}^{(k)} + \begin{bmatrix} \gamma\alpha^{(k+1)} \\ e^{(k+1)} \end{bmatrix} - \tilde{y}^{(k+1)}. \tag{63}$$

Note that with $\tilde{y}^{(0)} - \tilde{u}^{(0)} = 0$, we have $\alpha^{(1)} = \alpha_{\gamma,2}$, and the ADMM algorithm can then be interpreted as iterative "sparsification" of the ℓ_2-regularized "classical" linear prediction solution.

3.3. Solving a positive definite symmetric Toeplitz system

A classical algorithm for solving a (positive definite) symmetric Toeplitz system is the Levinson algorithm with time complexity $\mathcal{O}(N^2)$ and space complexity $\mathcal{O}(N)$ (Levinson, 1947; Golub and van Loan, 2013). Algorithms like the Levinson algorithm with time complexity $\mathcal{O}(N^2)$ are called fast algorithms, but there also exist superfast algorithms with time complexity $\mathcal{O}(N\log^2 N)$, see Bitmead and Anderson (1980) and Ammar and Gragg (1988). These algorithms also have the advantage that the first solution can be obtained in $\mathcal{O}(N\log^2 N)$ and any other solution with the same coefficient matrix but different right-hand side is possible with linearithmic time complexity $\mathcal{O}(N\log N)$. There are also algorithms where there is a one time penalty of $\mathcal{O}(N^2)$ and again all subsequent solutions with same coefficient matrix but a different right-hand side only requires $\mathcal{O}(N\log N)$ (Jain, 1979; Ammar and Gragg, 1988). The constant in the first step of a superfast algorithm is large and hence there is a break-even point in the number of operations at approximately $N = 256$ for N as a base 2 number (Ammar and Gragg, 1988). Since the experiments in Sections 4 and 5 uses $N < 256$ and that the $\mathcal{O}(N^2)$ algorithm in the first step is much simpler, we use the algorithm in Jain (1979), see also Ammar and Gragg (1988).

The inverse of a Toeplitz matrix $\bar{T}$ can be described by the Gohberg–Semencul formula

$$\delta_{N-1}\bar{T}^{-1} = \bar{T}_1\bar{T}_1^T - \bar{T}_0^T\bar{T}_0 \tag{64}$$

where

$$\bar{T}_0 = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \rho_0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \rho_{N-2} & \cdots & \rho_0 & 0 \end{bmatrix}, \tag{65}$$

and

$$\bar{T}_1 = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ \rho_{N-2} & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \rho_0 & \cdots & \rho_{N-2} & 1 \end{bmatrix}, \tag{66}$$

and

$$\delta_{N-1}\bar{T}^{-1} = \begin{bmatrix} 1 & \rho_{N-2} & \cdots & \rho_0 \\ \rho_{N-2} & & & \\ \vdots & & S & \\ \rho_0 & & & \end{bmatrix} \tag{67}$$

and $S \in \mathbb{R}^{(N-1)\times(N-1)}$ denotes the remaining submatrix. The variables δ_{N-1} and $\rho_0, \ldots, \rho_{N-2}$ can be obtained using the Szegö recursion in $\mathcal{O}(N^2)$ operations as a one time–cost per system. The solution to the system $\bar{T}x = b$ is then given by

$$x = \bar{T}^{-1}b = \frac{1}{\delta_{N-1}}\left(\bar{T}_1\bar{T}_1^T b - \bar{T}_0^T\bar{T}_0 b\right). \tag{68}$$

Since $\bar{T}_0$, $\bar{T}_1$ are Toeplitz, a product like $T_0 b$ can be evaluated via FFTs/IFFTs, see Jain (1979) for an algorithm for fast evaluation of (68) in $\mathcal{O}(N\log N)$ operations. So, all subsequent solutions with the same coefficient matrix are available in $\mathcal{O}(N\log N)$ operations. Recall that the coefficient matrix is $X^TX + I = R + I$ for the DR algorithm and $X^TX + \gamma^2 I = R + \gamma^2 I$ for the ADMM algorithm. From the perspective of signal processing, the coefficient matrices $X^TX + I = R + I$ and $X^TX + \gamma^2 I = R + \gamma^2 I$ are updated for each frame. It is only during each call of the DR and ADMM algorithms that the coefficient matrix is fixed. For each call the computation of δ_{N-1} and $\rho_0, \ldots, \rho_{N-2}$ with appropriate discrete Fourier transforms (DFTs) of the latter sequence can be seen as an initial caching to make subsequent iterations cheaper (as in, e.g., Afonso et al., 2010). Note that the diagonal term provides regularization to the solver.

4. Implementation and empirical computation time

The DR and ADMM algorithms for solving the SpLP problem using the Levinson algorithm to solve the symmetric positive definite Toeplitz system, are denoted **DR-L** and **ADMM-L**. The DR and ADMM algorithm that are using the method in Jain (1979) to solve the symmetric positive definite Toeplitz system are denoted **DR-GS** and **ADMM-GS**.

The algorithms[3] were implemented in C++ using Intel Math Kernel Library (MKL) (Intel(R) Math Kernel Library) for BLAS level 1 routines. The application of matrix–vector product with X and X^T was implemented as FFT filtering using the FFTW3 library (Frigo and Johnson, 2005). The time from call of the solver to return was measured using the POSIX function `gettimeofday` (for the C++ implementations). The timing was measured over 100 executions of each frame to average out possible system processes (note that each frame was then static and the solvers then run with the exact same input). The setting $\gamma = 0.12$ was found in Section 5 and fixed for all simulations. The simulations were executed on an Intel (R) Dual Core(TM) i5-2410M CPU at 2.3 GHz with Ubuntu Linux kernel 3.2.0–32-generic, MKL 10.3 and Matlab 7.13.0.564. The algorithms implemented with C++ were compiled using gcc-4.8 and the `-Os -March=native` optimization option. We compared the implementation with the general purpose software **Mosek** 7.0 (Andersen et al., 2003) and **CVX+SeDuMi** 1.21 (Grant and Boyd, 2011; Sturm, 1999). This problem was too large to use CVXGEN (Mattingley and Boyd, 2012). Algorithms **Cprimal** and **Cprimal(s/d)** are presented in Jensen et al. (2013) and were C++ implementations of interior-point methods.

Analytically motivated selections of parameters ρ, t requires that one of the functions applied to the Douglas–Rachford setup are smooth and/or strongly convex, see Patrinos et al. (2014). Since this is not the case, we empirically found $\eta = 1.8$ and $t = 0.1$ to be useful. For the ADMM based algorithms we empirically found $\rho = 100$ to be useful.

We chose the following stopping criteria. Algorithms **ADMM-L** and **ADMM-GS** were stopped if

$$\frac{1}{M+N}\|\tilde{z}^{(k+1)} - \tilde{y}^{(k+1)}\|_2^2 \leqslant \epsilon \tag{69}$$

$$\frac{\rho}{M+N}\|\tilde{y}^{(k+1)} - \tilde{y}^{(k)}\|_2^2 \leqslant \epsilon. \tag{70}$$

This reflected the primal and dual residual and was (up to a scaling of $\rho = 100$) the absolute criteria in Boyd et al. (2011). The algorithms **DR-L** and **DR-GS** were stopped if

$$\frac{1}{N+M}\|z^{(k)} - z^{(k-1)}\|_2^2 \leqslant \epsilon. \tag{71}$$

For both the DR and ADMM based algorithms we selected $\epsilon = 10^{-6}$ and also stopped if a maximum of $k = 100$ iterations were reached. Such a maximum of allowed iterations is useful to bound the worst-case execution time.

Here we presented results for $T = 320$, $N = 250$ ($M = 570$) processed on a 2 s speech signal sampled at 16 kHz taken from the testing database. The results are shown in Table 1.

From Table 1 we observed that the splittings methods are orders faster than general purpose software and one order faster than hand-tailored interior-point methods. It was also clear that using the algorithm (Jain, 1979) for solving the auxiliary symmetric positive definite Toeplitz system was faster that the classical Levinson–Durbin algorithm for these dimensions (otherwise the methods were equivalent in that the same iterations are generated).

[3] MATLAB and C++ implementations are available from http://kom.aau.dk/~tlj/.

Table 1
Timing in milliseconds. Format: **Average**/min/max. The setting is $T = 320$, $N = 250$ ($M = 570$).

Methods	Timings
CVX + SeDuMi	2467.29/1327.29/3619.74
Mosek	224.71/145.54/307.60
Cprimal	92.70/55.24/180.46
Cprimal(s/d)	63.66/33.59/112.09
DR-L	6.62/0.65/10.11
DR-GS	2.28/0.61/3.26
ADMM-L	2.99/0.65/5.14
ADMM-GS	1.29/0.61/1.92

The ADMM based methods converged faster to the used stopping criteria and this was the reason for being faster than the DR based methods. Specifically, the ADMM based algorithms used on average 13.5 iterations, while the DR based algorithms used 35.3 iterations.

The splitting methods solved the problem to a low accuracy. Specifically, using the metrics

$$m_{\mathbf{DR}} = \frac{f_{\mathbf{DR}} - f^\star}{f^\star}, \qquad m_{\mathbf{ADMM}} = \frac{f_{\mathbf{ADMM}} - f^\star}{f^\star} \tag{72}$$

we observed than on average $m_{\mathbf{DR}}$ and $m_{\mathbf{ADMM}}$ is 0.14 and 0.12, respectively, i.e. approximately only a 10^{-1} sub-optimal solution ($f^\star$ is approximated via the solution from **Mosek**). The impact of this low accuracy was then assessed in the experimental analysis, presented in Section 5. An example of the different convergence behaviors of the DR and ADMM algorithms is illustrated in Fig. 1 with the metric $\frac{f(\alpha^{(k)})-f^\star}{f^\star}$ using a single frame from the simulations in Table 1. Notice that the **ADMM-L** algorithm converges faster the few first iterations. The endpoint of the graphs illustrates where the stopping criteria was activated and stopped the iterative algorithms.

Algorithms **CVX+SeDuMi**, **Mosek**, **Cprimal**, **Cprimal(s/d)** are all primal–dual interior-point methods, and the accuracy of these methods is all higher than using the DR and ADMM algorithms. However, the DR and ADMM algorithms have the advantage that some of the elements in an approximate solution $\hat{\alpha} \approx \alpha^\star$ are exactly 0 due to the soft-thresholding function (25) applied at (21) and (62). On the other-hand, interior-point methods reformulate the problem as a constrained problem and approach the solution from the interior but never exactly reach the bound of the constraints of the reformulated problem, such that the small magnitude elements in $\hat{\alpha}_{\mathrm{IP}} \approx \alpha^\star$ are small but not exactly 0.

5. Experimental analysis

In this section, we outline some of the properties of the different LP models introduced in Section 2 and, in particular the SpLP method of Section 2.5 equipped with the solvers proposed in Section 3. Firstly, we present their objective spectral modeling properties by analyzing their prediction gain. Secondly, we provide a more practical example of their modeling properties in a speech reconstruction scenario where we also evaluate the goodness of the approximate solutions of Section 3 through perceptual objective quality measures. The algorithms compared are outlined in Table 2. The splitting methods solvers in algorithms **HOSpLPdr** and **HOSpLPadmm** used the same stopping criteria as outlined in Section 4. The TIMIT database (Garofolo et al., 1993) was chosen for the analysis because of its manageable size with a sufficiently large number of speakers for testing speech processing algorithms accuracy. Since our algorithms windows are in the order of few milliseconds, we can reasonably assume that the conclusions of the experimental analysis done with the TIMIT dataset extent to a larger class of speakers.

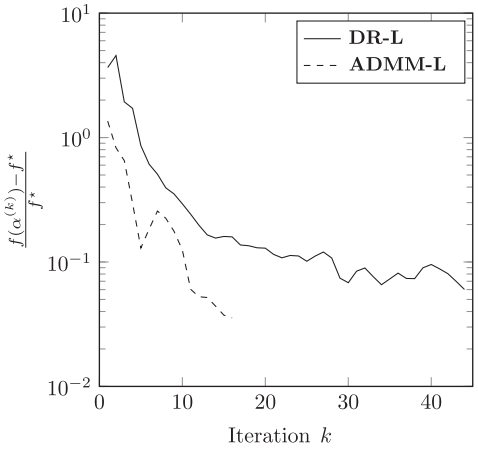

Fig. 1. Example of the convergence behavior of the algorithms **DR-L** and **ADMM-L**. The endpoints of the graphs illustrates where the stopping criteria has become active and stopped the iterative algorithm.

5.1. Prediction gain

The vowel and semivowel phones (Halberstadt and Glass, 1997) from the TIMIT database (sampled at 16 kHz) were processed, belonging to 3696 sentences from 462 speakers. We chose the ones of duration of at least 640 samples (40 ms) for a total of about 40,000 voiced speech frames.

The methods compared are presented in Table 2. In **LTP**, **LTP3**, and **LTP3j** the short-term predictor had 20 coefficients. For the purposes of comparison, the autocorrelation method was used for both the short-term and long-term predictors. The value of the pitch lag in **LTP** and **LTP3** was chosen such that the prediction gain was maximized. This was accomplished by an exhaustive search over the allowable range $T_p \in [34, 231]$, as used in AMR-WB (Bessette et al., 2002), effectively covering pitch frequencies belonging to the range $[69, 470]$ Hz. For **LTP3j**, the pitch lag found in **LTP3** was used in the optimization. Note that, while a plethora of methods exist for pitch estimation (see, e.g., Christensen and Jakobsson, 2009), we chose the exhaustive search to guarantee the prediction

Table 2
Prediction methods used for comparison.

Method	Description
LTP	Combined 20 tap short-term LP and 1 tap long-term LP calculated separately
LTP3	Combined 20 tap short-term LP and 3 tap long-term LP calculated separately
LTP3j	Combined 20 tap short-term LP and 3 tap long-term LP calculated jointly using (11)
HOLP	High-order LP with 2-norm criterion calculated with high-order (Section 2.4)
HOSpLPip	High-order sparse LP (14) obtained with interior point solution
HOSpLPdr	High-order sparse LP (14) obtained with DR algorithm
HOSpLPadmm	High-order sparse LP (14) obtained with ADMM algorithm

performance not to be biased by a possible erroneous estimation of the pitch lag.

In the methods **HOLP**, **HOSpLP**, **HOSpLPdr** and **HOSpLPadmm** the order of the filter was fixed to 250, allowing to cover the above-mentioned pitch lag range and a related bulk of nonzero coefficients clustered around the maximum allowable pitch lag. In order to find an appropriate value of the regularization parameter, we analyzed a set of voiced speech frames (different from the one used in the experiments) and determined the point of maximum curvature on the curve $(\|\alpha\|_1, \|x - X\alpha\|_1)$, a modified version of the L-curve (Hansen and O'Leary, 1993), appropriate to determine trade-offs between the sparsity of the predictor and the sparsity of the residual. The regularization parameter was then chosen fixed, $\gamma = 0.12$.

For a fair comparison, after calculating **HOSpLP**, **HOSpLPdr** and **HOSpLPadmm**, only the 21 largest values were retained, this kept the actual sparsity of the methods the same as the simple **LTP**. Note that the actual number of nonzero samples for **LTP** would be 40 considering the convolution of short-term and long-term predictors in the analysis.

The average prediction gains are shown in Table 3. The 95% confidence intervals showed a clear distinction between the various method as well as a certain consistency in performance. Clearly, **HOLP** was the best performing method. This behavior can be explained from Parsevals theorem and the power spectrum matching properties of the all-pole spectrum obtained with 2-norm LP that could approximate the power spectrum of a signal with arbitrarily small error (Makhoul, 1975). The performance of **HOLP** then determined the upper bound of performance.

Table 3
Average prediction gains for segments of different length N. A 95% confidence interval is shown. The number of nonzero elements, **card**$(\cdot)$, is shown for comparison.

METHOD	**Card**$(\cdot)$	T	
		320	640
LTP	21	17.3 ± 0.8	14.2 ± 1.0
LTP3	23	22.3 ± 0.8	19.9 ± 0.9
LTP3j	43	24.2 ± 0.6	22.6 ± 0.8
HOLP	250	32.4 ± 0.6	31.3 ± 0.7
HOSpLPip	21	28.6 ± 1.1	27.8 ± 1.4
HOSpLPdr	21	28.5 ± 1.4	27.6 ± 1.6
HOSpLPadmm	21	28.3 ± 1.7	27.2 ± 1.6

The methods **HOSpLP**, **HOSpLPdr** and **HOSpLPadmm** behaved, in statistical terms, identically, thus providing further proof of the reliability of the proposed fast solution, even though the latter two calculated a much less accurate solution to the problem (14). In general, the sparseness criterion helped providing a net reduction in the number of nonzero samples while obtaining just slightly lower performance. The **LTP**, **LTP3**, and **LTP3j** methods provided proof of the gain in prediction gain given by the joint estimation of short-term and long-term coefficients provided by the high-order sparse model which achieves more than 10 dB gain compared to traditional **LTP**.

A proof of concept example is shown in Figs. 2 and 3 for a 640 samples segment of the vowel/a/uttered by a female speaker where the predictors and related frequency magnitudes are shown. Visually, the dissimilarities between **LTP3j** and the sparse methods, which behave very similarly, come mostly from the lower order short-term predictor necessary to model the envelope (around 10 taps versus 20) and the larger cluster of taps around T_p needed to model the pitch redundancies. This allowed the sparse methods not to just have a more parsimonious representation, but also allowed for general better modeling thanks to

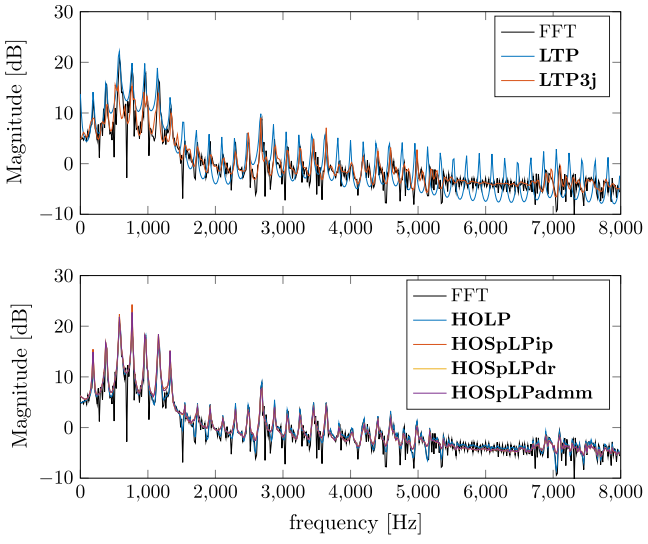

Fig. 2. Magnitude of the frequency response of the different methods proposed. A 640 samples segment of the voiced speech (vowel/a/uttered by a female speaker) is used for the analysis.

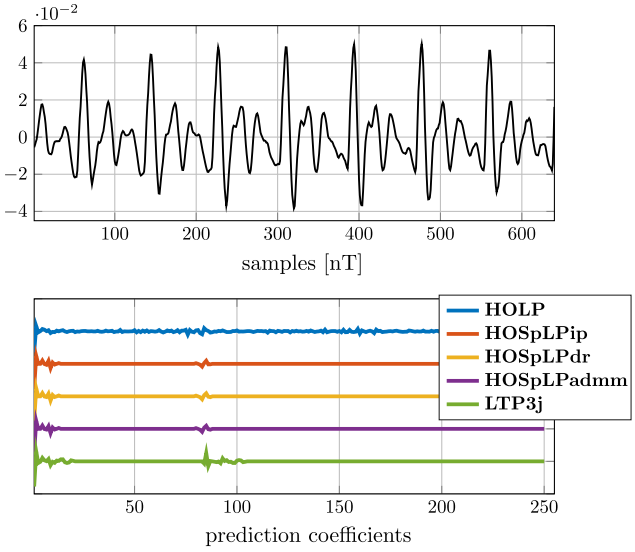

Fig. 3. A 640 samples segment of the voiced speech (vowel/a/uttered by a female speaker) and the calculated predictors for the different methods proposed.

the sparse criteria imposed both on the predictor and residual.

5.2. Speech reconstruction

Considering speech as a slowly-varying process, we can verify the performance of the different predictors presented in Table 2 in a statistical cross-validation framework. In particular, for a given speech segment, we can determine how accurately the predictors perform in reconstructing a set of unknown speech samples given the known samples and their prediction model. The problem can then be rewritten as a maximum a posteriori (MAP) estimator

$$\underset{x_u}{\text{maximize}} \quad p(x_u|x_k,\alpha) \tag{73}$$

where x_k represents the known samples and x_u represents the unknowns samples of a given speech segment of length T and the predictor α is calculated using only the known samples. It is therefore imperative that α is effective in modeling the underlying statistics without underfitting or overfitting in order to have the best estimate of x_u. This problem is well-known in the statistical audio processing literature as AR model-based speech reconstruction (Godsill and Rayner, 1998). Considering the segment of T speech samples x as partitioned in terms of known and unknown samples

$$x = Kx_k + Ux_u, \tag{74}$$

where U and K are $T \times T$ "rearrangements" matrices that form a columnwise partition of the identity matrix I, and if we consider the data samples x as drawn by an AR process with parameters α, we can rewrite the interpolation error as

$$e = A(Kx_k + Ux_u) \tag{75}$$

where A is the so-called analysis matrix obtained with α (Godsill and Rayner, 1998), thus (75) is just another way to rewrite the system of equations in (2). If we fit the interpolation error into an i.i.d. Gaussian process, which is reasonable given the limited knowledge of the reconstruction process, we obtain

$$\begin{aligned} p(x_u|x_k,\alpha) &\propto \exp(-\|e\|_2^2) \\ &= \exp(-\|A(Kx_k + Ux_u)\|_2^2). \end{aligned} \tag{76}$$

Maximizing the argument of (76), we obtain

$$x_u = -\left(A_u^T A_u\right)^{-1} A_u^T A_k x_k \tag{77}$$

where $A_u = AU$ and $A_k = AK$. This solution is then equivalent to minimizing the mean-square error of e of (75). Note that the basic formulation given in (73) assumes that the AR parameters are known *a priori*. In practice, there are ways to obtain a robust estimate during the detection stage (Godsill and Rayner, 1998), however, in our case we will limit ourselves to estimating α over the known speech samples. Considering the reconstruction Eq. (77), it can be seen that traditional LP model might fail to properly reconstruct the pitch periodicity if the estimator of the model parameters α used to generate A do not account for long-term redundancies.

We compared the different methods presented in Table 2 in the reconstruction approach presented in (73) to estimate the predictor used to generate the matrix A. Differently from Section 5.1, in this section we measured the reconstruction using the mean opinion score (MOS), as calculated through POLQA (ITU-T, 2010), to account for perceptual qualities as well.

In this experiment we targeted both voiced and unvoiced speech, in order to provide proof of the overall robustness of the sparse linear predictor introduced. This is different from Section 5.1, where we targeted uniquely voiced speech as a proof of concept. The comparison was carried out for missing segment length T_{gap} of 4, 6, 8, 10, and 20 ms, respectively, 64, 96, 128, 160, and 320 samples at 16 kHz. We process 1000 sentences coming from several different speakers with different characteristics (gender, age, pitch, regional accent) taken from the TIMIT database. We applied a robust speech activity detector to avoid applying the reconstruction and calculating statistics over silence. For each file, the losses were produced every 150 ms, the 40 ms (640 samples) before the loss are used to generate the known vector x_k, while the varying length gap forms the unknown vector x_u. The predictor was calculated with the methods presented in Table 2 on the known vector x_k, the unknown segment was then reconstructed using (77). For comparison, we have also added a traditional low-order method **sLP** of order 20.

The results shown in Table 4 gave a different perspective on the performance of the different predictors. While the prediction gain results of Table 3 showed **HOLP** to perform significantly better, in terms of perceptual quality of the reconstructed signal the higher order did not mean

Table 4
Average MOS for speech reconstruction with different gap size losses. The 40 ms before the loss are known and used in the reconstruction framework (77). A 95% confidence interval is shown.

METHOD	T_{GAP} [ms]				
	4	6	8	10	20
sLP	3.92 ± 0.09	3.15 ± 0.15	2.96 ± 0.16	2.30 ± 0.18	1.71 ± 0.22
LTP	4.13 ± 0.07	3.44 ± 0.14	3.17 ± 0.12	2.71 ± 0.09	2.45 ± 0.13
LTP3	4.17 ± 0.07	3.53 ± 0.09	3.22 ± 0.13	2.92 ± 0.12	2.63 ± 0.09
LTPj	4.12 ± 0.05	3.63 ± 0.12	3.31 ± 0.12	3.00 ± 0.11	2.75 ± 0.16
HOLP	4.27 ± 0.04	3.55 ± 0.06	3.34 ± 0.08	2.91 ± 0.09	2.61 ± 0.11
HOSpLPip	4.34 ± 0.03	3.75 ± 0.05	3.56 ± 0.08	3.27 ± 0.09	3.12 ± 0.15
HOSpLPdr	4.34 ± 0.02	3.74 ± 0.08	3.55 ± 0.07	3.27 ± 0.11	3.12 ± 0.12
HOSpLPadmm	4.31 ± 0.04	3.69 ± 0.07	3.54 ± 0.07	3.24 ± 0.08	3.11 ± 0.11

higher quality, unless it has actually a clear meaning of representing short-term and long-term redundancies of the speech signal. Thus, **HOSpLP**, **HOSpLPdr** and **HOSpLPadmm** performed better being more accurate and avoiding overfitting the data. It was interesting to notice that **LTP3** and **LTP3j** also performed fairly closely to **HOLP**. The **sLP** method performance were close to the other 2-norm based methods for the smallest gap size, however they decay quite rapidly as the gap size increased. Finally, we noted that, differently from other interpolation approaches that involve a Estimation-Maximization (EM) approach to enhance the estimation of the AR model and reconstructed signal, while a slight increase in mean-squared error was achieved, no improvement in MOS was actually seen.

6. Discussion and conclusions

We presented algorithms suitable for finding approximate solutions to the high-order sparse linear prediction problem in speech applications. In particular, we pointed out that a lower accuracy and slower convergence did not affect the overall performance of the predictors when applied in realistic applications that required both objective and subjective quality metrics to be met. The resilience to this approximation could be explained from the actual solution needed by the problem being actually different from the "true" 1-norm solution found by the interior point method. We are indeed looking for a residual and predictor that are "small" and with sparse structures and not particularly for the 1-norm solution. Thus, further work can include better sparse approximations, rather than seeking more accurate convergence methodologies, as done with, e.g., interior point methods.

Acknowledgement

The authors would like to thank Prof. Søren Holdt Jensen for pointing them to superfast solvers for Toeplitz systems.

References

Afonso, M., Bioucas-Dias, J., Figueiredo, M., 2010. Fast image recovery using variable splitting and constrained optimization. IEEE Trans. Image Process. 19 (9), 2345–2356.

Alipoor, G., Savoji, M.H., 2012. Wide-band speech coding based on bandwidth extension and sparse linear prediction. In: Proc. International Conference on Telecommunications and Signal Processing, pp. 454–459.

Ammar, G., Gragg, W., 1988. Superfast solution of real positive definite Toeplitz systems. SIAM J. Matrix Anal. Appl. 9 (1), 61–76.

Andersen, E.D., Roos, C., Terlaky, T., 2003. On implementing a primal-dual interior-point method for conic quadratic optimization. Math. Program. Series B 95 (2), 249–277.

Angelosante, D., 2014. Sparse regressions for joint segmentation and linear prediction. In: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 335–339.

Angelosante, D., Giannakis, G., Sidiropoulos, N., 2013. Sparse parametric models for robust nonstationary signal analysis: leveraging the power of sparse regression. IEEE Signal Process. Mag. 30 (6), 64–73.

Atal, B.S., 2006. The history of linear prediction. IEEE Signal Process. Mag. 23 (2), 154–161.

Atal, B.S., Hanauer, S.L., 1971. Speech analysis and synthesis by linear prediction of the speech wave. J. Acoust. Soc. Am. 50 (2B), 637–655.

Bäckström, T., 2004. Linear predictive modelling of speech: constraints and line spectrum pair decomposition. Doctoral Dissertation. Helsinki University of Technology.

Beck, A., Teboulle, M., 2009. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imag. Sci. 2 (1), 183–202.

Bensaid, S., Slock, D., 2012. Blind audio source separation exploiting periodicity and spectral envelopes. In: Proc. of International Workshop on Acoustic Signal Enhancement, pp. 1–4.

Bessette, B., Salami, R., Lefebvre, R., Jelinek, M., Rotola-Pukkila, J., Vainio, J.... Jarvinen, K., 2002. The adaptive multirate wideband speech codec (AMR-WB). IEEE Trans. Speech Audio Process. 10 (8), 620–636.

Bitmead, R., Anderson, B., 1980. Asymptotically fast solution of Toeplitz and related systems of linear equations. Linear Algebra Appl. 34, 103–116.

Bochud, N., Gomez, A., Rus, G., Peinado, A., 2013. Sparse signal model for ultrasonic nondestructive evaluation of CFRP composite plates. In: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 2844–2847.

Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends Mach. Learn. 3 (1), 1–122.

Christensen, M.G., Jakobsson, A., 2009. Multi-pitch estimation. Synthesis Lectures on Speech & Audio Processing. Morgan & Claypool Publishers.

Combettes, P.L., Pesquet, J.-C., 2007. A Douglas–Rachford splitting approach to nonsmooth convex variational signal recovery. IEEE J. Sel. Top. Signal Process. 1 (4), 564–574.

Combettes, P.L., Wajs, V.R., 2005. Signal recovery by proximal forward-backward splitting. Multiscale Model. Simul. 4, 1168–1200.

Defraene, B., van Waterschoot, T., Ferreau, H.J., Diehl, M., Moonen, M., 2012. Real-time perception-based clipping of audio signals using

convex optimization. IEEE Trans. Speech Audio Lang. Process. 20 (10), 2657–2671.
Defraene, B., van Waterschoot, T., Diehl, M., Moonen, M., 2014. Embedded-optimization-based loudspeaker precompensation using a Hammerstein loudspeaker model. IEEE/ACM Trans. Speech Audio Lang. Process. 22 (11), 1648–1659.
Denoel, E., Solvay, J.-P., 1985. Linear prediction of speech with a least absolute error criterion. IEEE Trans. Acoust. Speech Signal Process. 33 (6), 1397–1403.
Donoho, D.L., Elad, M., 2002. Optimally sparse representation in general (nonorthogonal) dictionaries via $\ell 1$ minimization. Proc. Natl. Acad. Sci. U.S.A. 4 (5), 2197–2202.
Douglas, J., Rachford, H.H., 1956. On the numerical solution of heat conduction problems in two and three space variables. Trans. Am. Math. Soc. 82, 421–439.
Eckstein, J., Bertsekas, D., 1992. On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators. Math. Program. 55 (1–3), 293–318.
Ekman, L.A., Kleijn, W.B., Murthi, M.N., 2008. Regularized linear prediction of speech. IEEE Trans. Speech Audio Lang. Process. 16 (1), 65–73.
El-Jaroudi, A., Makhoul, J., 1991. Discrete all-pole modeling. IEEE Trans. Signal Process. 39 (2), 411–423.
Erer, I., Sarikaya, K., Bozkurt, H., 2014. Enhanced radar imaging via sparsity regularized 2D linear prediction. In: Proc. European Signal Processing Conference, pp. 1751–1755.
Frigo, M., Johnson, S.G., 2005. The design and implementation of FFTW3. Proc. IEEE 93 (2), 216–231.
Gabay, D., 1983. Applications of the method of multipliers to variational inequalities. In: Fortin, M., Glowinski, F. (Eds.), Augmented Lagrangian methods: Applications to the Numerical Solution of Boundary-Value Problems, . In: Studies in Mathematics and Its Applications. Springer-Verlag, pp. 299–340.
Gabay, D., Mercier, B., 1976. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. Comput. Math. Appl. 2 (1), 17–40.
Garofolo, J., Lamel, L., Fisher, W., Fiscus, J., Pallett, D., Dahlgren, N., Zue, V., 1993. TIMIT Acoustic-Phonetic Continuous Speech Corpus LDC93S1. Web Download. Linguistic Data Consortium, Philadelphia.
Gersho, A., Gray, R.M., 1992. Vector Quantization and Signal Compression. Springer.
Giacobello, D., Christensen, M.G., Dahl, J., Jensen, S.H., Moonen, M., 2008. Sparse linear predictors for speech processing. In: Proc. of the Annual Conference of the International Speech Communication Association, pp. 1353–1356.
Giacobello, D., Christensen, M.G., Dahl, J., Jensen, S.H., Moonen, M., 2009. Joint estimation of short-term and long-term predictors in speech coders. In: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 4109–4112.
Giacobello, D., Christensen, M.G., Murthi, M.N., Jensen, S.H., Moonen, M., 2009. Speech coding based on sparse linear prediction. In: Proc. European Signal Processing Conference, pp. 2524–2528.
Giacobello, D., Christensen, M.G., Murthi, M.N., Jensen, S.H., Moonen, M., 2012. Sparse linear prediction and its applications to speech processing. IEEE Trans. Speech Audio Lang. Process. 20 (5), 1644–1657.
Glowinski, R., Marroco, A., 1975. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problémes de Dirichlet non linéaires. ESAIM: Math. Model. Numer. Anal. – Modél. Math. Anal. Numér. 9 (R2), 41–76.
Godsill, S., Rayner, P., 1998. Digital Audio Restoration. Springer.
Golub, G.H., van Loan, C.F., 2013. Matrix Computations, fourth ed. Johns Hopkins University Press.
Grant, M., Boyd, S., 2011. CVX: Matlab Software for Disciplined Convex Programming, Version 1.21. <http://cvxr.com/cvx/>.
Halberstadt, A.K., Glass, J.R., 1997. Heterogeneous acoustic measurements for phonetic classification. In: Proc. European Conference on Speech Communication and Technology, pp. 401–404.
Hansen, P.C., O'Leary, D.P., 1993. The use of the L-curve in the regularization of discrete ill-posed problems. SIAM J. Sci. Comput. 14 (6), 1487–1503.
Hansen, J.H.L., Proakis, J.G., Deller Jr., J.R., 1987. Discrete-Time Processing of Speech Signals. Prentice Hall.
Hermansky, H., 1990. Perceptual linear predictive (PLP) analysis of speech. J. Acoust. Soc. Am. 87 (4), 1738–1752.
Intel(R) Math Kernel Library. <https://software.intel.com/en-us/intel-mkl>.
Itakura, F., Saito, S., 1970. A statistical method for estimation of speech spectral density and formant frequencies. Electron. Commun. Jpn. 53 (A), 36–43.
ITU-T, 2010. Perceptual Objective Listening Quality Assessment, p.863.
Jain, J.R., 1979. An efficient algorithm for a large Toeplitz set of linear equations. IEEE Trans. Acoust. Speech Signal Process. 27 (6), 612–615.
Jensen, T., Giacobello, D., Christensen, M., Jensen, S., Moonen, M., 2013. Real-time implementations of sparse linear prediction for speech processing. In: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 8184–8188.
Kabal, P., Ramachandran, R., 1989. Joint optimization of linear predictors in speech. IEEE Trans. Acoust. Speech Signal Process. 37 (5), 642–650.
Kameoka, H., Ono, N., Sagayama, S., 2010. Speech spectrum modeling for joint estimation of spectral envelope and fundamental frequency. IEEE Trans. Speech Audio Lang. Process. 18 (6), 1507–1516.
Kay, S.M., 1979. The effects of noise on the autoregressive spectral estimator. IEEE Trans. Acoust. Speech Signal Process. 27 (5), 478–485.
Koloda, J., Østergaard, J., Jensen, S.H., Sánchez, V.E., Peinado, A.M., 2013. Sequential error concealment for video/images by sparse linear prediction. IEEE Trans. Multimedia 15 (4), 957–969.
Lee, C.-H., 1988. On robust linear prediction of speech. IEEE Trans. Acoust. Speech Signal Process. 36 (5), 642–650.
Levinson, N., 1947. The Weiner RMS error criterion in filter design and prediction. J. Math. Phys. MIT 25, 261–278.
Lions, P.L., Mercier, B., 1979. Splitting algorithms for the sum of two nonlinear operators. SIAM J. Numer. Anal. 16 (6), 964–979.
Magi, C., Pohjalainen, J., Bäckström, T., Alku, P., 2009. Stabilised weighted linear prediction. Speech Commun. 51 (5), 401–411.
Makhoul, J., 1975. Linear prediction: a tutorial review. Proc. IEEE 63 (4), 561–580.
Mattingley, J., Boyd, S., 2010. Real-time convex optimization in signal processing. IEEE Signal Process. Mag. 27 (3), 50–61.
Mattingley, J., Boyd, S., 2012. CVXGEN: a code generator for embedded convex optimization. Optim. Eng. 13 (1), 1–27.
Moreau, J.J., 1965. Proximité et dualité dans un espace hilbertien. Bull. Soc. Math. France 93, 273–299.
Murthi, M., Rao, B., 1998. Towards a synergistic multistage speech coder. In: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. I/369–I/372.
Murthi, M.N., Rao, B.D., 2000. All-pole modeling of speech based on the minimum variance distortionless response spectrum. IEEE Trans. Speech Audio Process. 8 (3), 221–239.
Nesterov, Y., 2007. Gradient methods for minimizing composite objective function, Center for Operations Research and Econometrics (CORE) Discussion Papers no. 2007/76, Université Catholique de Louvain.
Nesterov, Y., Nemirovskii, A., 1994. Interior-Point Polynomial Methods in Convex Programming. SIAM.
O'Connor, D., Vandenberghe, L., 2014. Primal-dual decomposition by operator splitting and applications to image deblurring. SIAM J. Imag. Sci. 7 (3), 1724–1754.

Patrinos, P., Stella, L., Bemporad, A., 2014. Douglas–Rachford splitting: complexity estimates and accelerated variants. In: Proc. IEEE Conference on Decision and Control, pp. 4234–4239.

Saito, S., Itakura, F., 1967. Theoretical consideration of the statistical optimum recognition of the spectral density of speech. J. Acoust. Soc. Jpn.

Spingarn, J., 1985. Applications of the method of partial inverses to convex programming: decomposition. Math. Program. 32 (2), 199–223.

Stoica, P., Moses, R.L., 2005. Spectral Analysis of Signals. Prentice Hall.

Sturm, J.F., 1999. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. Optim. Methods Software 11–12, 625–653.

Thyssen, J., Nielsen, H., Hansen, S.D., 1994. Non-linear short-term prediction in speech coding. In: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. I/185–I/188.

Vaidyanathan, P.P., 2009. The theory of linear prediction. Synthesis Lectures on Signal Processing. Morgan & Claypool Publishers.

Vandenberghe, L., 2013. Douglas–Rachford Method and ADMM, Optimization Methods for Large-Scale Systems, Lecture Notes, University of California, Los Angeles (UCLA).

van Waterschoot, T., Moonen, M., 2008. Comparison of linear prediction models for audio signals. EURASIP J. Audio Speech Music Process.

Wright, S.J., Nowak, R., Figueiredo, M.A.T., 2009. Sparse reconstruction by separable approximation. IEEE Trans. Signal Process. 57 (7), 2479–2493.

Yang, J., Zhang, Y., 2011. Alternating direction algorithms for ℓ_1-problems in compressive sensing. SIAM J. Sci. Comput. 33 (1), 250–278.

Yin, W., Osher, S., Goldfarb, D., Darbon, J., 2008. Bregman iterative algorithms for ℓ_1-minimization with application to compressed sensing. SIAM J. Imag. Sci. 1 (1), 143–168.