TWO-LEVEL OPTIMIZATION APPROACH WITH ACCELERATED PROXIMAL GRADIENT FOR OBJECTIVE MEASURES IN SPARSE SPEECH RECONSTRUCTION

HAI HUYEN DAM
School of Electrical Engineering, Computing and Mathematical Sciences
Curtin University of Technology, Perth, Australia

SIOW YONG LOW
School of Electronics and Computer Science
University of Southampton, Malaysia

SVEN NORDHOLM
School of Electrical Engineering, Computing and Mathematical Sciences
Curtin University of Technology, Perth, Australia

(Communicated by Changzhi Wu)

Abstract. Compressive speech enhancement makes use of the sparseness of speech and the non-sparseness of noise in time-frequency representation to perform speech enhancement. However, reconstructing the sparsest output may not necessarily translate to a good enhanced speech signal as speech distortion may be at risk. This paper proposes a two level optimization approach to incorporate objective quality measures in compressive speech enhancement. The proposed method combines the accelerated proximal gradient approach and a global one dimensional optimization method to solve the sparse reconstruction. By incorporating objective quality measures in the optimization process, the reconstructed output is not only sparse but also maintains the highest objective quality score possible. In other words, the sparse speech reconstruction process is now quality sparse speech reconstruction. Experimental results in a compressive speech enhancement consistently show score improvement in objectives measures in different noisy environments compared to the non-optimized method. Additionally, the proposed optimization yields a higher convergence rate with a lower computational complexity compared to the existing methods.

1. Introduction. In many practical applications with speech interfaces, speech signal is invariably degraded by background noise. A solution to noisy speech is to perform speech enhancement so that the overall perceptual quality and the intelligibility of the degraded speech is improved. Typically, both the noise estimation and suppression are performed in the frequency domain due to its computational simplicity and relative effectiveness [2, 3]. However, the problem becomes complicated when the background noise is highly non-stationary e.g., restaurant or cafeteria noise [11]. Numerous noise estimators have been proposed with varying
results [17, 2]. As speech is also non-stationary, analyzing the overlapped spectral components of speech and noise can be challenging. Apart from the performance degradation due to erroneous noise estimation, the noise mismatch will result in the infamous “musical noise” or noise artefacts [10, 18, 26].

Recently, compressive speech enhancement (CSE) has been employed as a speech enhancement method which bypasses the need for noise estimation. CSE primarily exploits the sparsity of speech by making use of the fact that speech is sparse in the time-frequency domain whilst noise is not [16, 15, 27]. Coupled with compressed sensing theory, which states that sparse signals with a small set of measurements can be reconstructed with an overwhelming probability [5], then given a mixture of sparse and non-sparse signals to be reconstructed, only sparse signal (i.e., speech signal) will be reconstructed [6, 7]. As such, sparse reconstruction is effectively transformed into a speech enhancement method in the time-frequency domain.

One of the most well known sparse signal reconstruction method is the regularized $\ell_1$-norm least squares [25, 22]. It has been shown that the $\ell_1$ regularized least squares yields a sparser solution as the solution tends to have a fewer non-zero coefficients compared to $\ell_2$ based Tikhonov regularization [12, 28]. A critical parameter in solving the regularized sparse solution is the regularization parameter, $\lambda$ as it is the key parameter in setting how sparse a solution is reconstructed. Kim et al. [12] pointed out that $\lambda$ provides a reasonable trade-off between the smoothness of the reconstructed signal and similarity to the original signal. Ideally, the smoothness should match the sparsity of the desired signal in question for good reconstruction. Evidently, $\lambda$ has an important role to play in speech sparse reconstruction as it regulates the sparsity level of the reconstructed signal.

As the sparsest solution may not necessarily translate to an improvement in the overall speech quality, the sparsity level of the output should match closely to the desired signal [23]. In fact, as far as speech enhancement applications are concerned, the sparsity level can be viewed as the noise suppression level. By setting the solution too sparse, the noise suppression will be at a high level, which potentially could lead to speech distortion. Clearly, there is a need to set the sparsity level of the reconstructed signal such that the overall speech quality is improved. This paper proposes to address this research question by incorporating speech quality measures into the sparse reconstruction process through the regulation of $\lambda$. This can be achieved by optimizing $\lambda$ to closely match the sparsity level of the reconstructed signal to that of the desired signal, thereby improving the overall speech quality. As such, the sparse speech reconstruction process effectively reconstructs sparse speech solution, which also improves the speech quality.

First, we investigate different optimization methods for solving the sparse reconstruction optimization problem with a fixed value of $\lambda$. We propose the use of the proximal gradient method [20]–[19] to solve the optimization problem to reduce the computational complexity when compared with existing method such as the interior point method [12]. The proximal gradient method is simple to implement in practice as it requires only the calculation of the gradient vector for each iteration. Secondly, we propose to combine the accelerated method [1] with the proximal gradient method for further reducing the complexity associated with the gradient method and for improving the convergence rate.

The problem of optimizing the regularization parameter, $\lambda$, is investigated further where we use a two-level optimization approach for solving the problem. In this case, one dimensional optimization process where objective quality measures (such as
the perceptual evaluation of speech quality (PESQ) [21] or the short-time objective intelligibility (STOI) [24]) serve as the cost function is combined with the accelerated proximal gradient method. The end result is an optimized regularization parameter, \( \lambda \), which yields a sparse solution with the highest score possible for the objective measures. Experimental results in noisy settings show that the optimized \( \lambda \) achieves a higher score in both PESQ and STOI compared to the non-optimized approach. The results also show that proposed accelerated proximal gradient method obtains a higher convergence rate with a lower computational requirement when compared with existing method using the proximal gradient method or the interior point method [12].

The paper is organized as follows. The problem formulation is formulated in Section 2. The proximal gradient method and the accelerated gradient method with updated step size are discussed in Section 3. The optimization problem with respect to the regularization parameter \( \lambda \) is investigated in Section 4. Simulation results are given in Section 5 and finally concluding remarks are in Section 6.

2. Problem formulation. Let the noisy signal be

\[
x(n) = s(n) + v(n)
\]

where \( s(n) \) and \( v(n) \) are the speech and noise signals, respectively. The short-time frequency representation of the noisy signal is given as

\[
X(\omega, k) = \sum_{n=0}^{L-1} x(n)w(n-kR)e^{-j\omega n} = S(\omega, k) + V(\omega, k)
\]

where \( w(n-kR) \) is a length \( L \) window function with a hop size of \( R \), \( k \) is the time index and \( \omega \in \{\omega_0, \cdots, \omega_{L-1}\} \). The \( k \)-th instant data envelope of (2) is \(|X(\omega, k)|\), where \(| \cdot |\) denotes the absolute value operator. The \( N \) length envelope vector of the noisy signal is \( \mathbf{x}(\omega, k) = [|X(\omega, k)|, |X(\omega, k-1)|, \cdots, |X(\omega, k-N+2)|, |X(\omega, k-N+1)|]^T \) and the symbol \(| \cdot |^T \) is the transposition operator.

Denote by \( \mathbf{Ψ} \) an \( N \times N \) matrix whose columns form an orthonormal basis. The \( K \)-sparse signal, \( \mathbf{x}(\omega, k) \in \mathbb{R}^N \) can be expressed as

\[
\mathbf{x}(\omega, k) = \mathbf{Ψ}(\omega)\theta(\omega, k)
\]

where \( \theta(\omega, k) \in \mathbb{R}^N \) has \( K \) non-zero entries. From the theory of compressed sensing, the measurement vector is

\[
\mathbf{y}(\omega, k) = \Phi(\omega)\mathbf{x}(\omega, k)
\]

where \( \Phi(\omega) \) is a \( M \times N \) sensing matrix. The role of the sensing matrix is to compress the signal’s envelope for each frequency \( \omega \) by making the measurements as incoherent as possible to reduce redundancy. Due to the compression capability, equation (4) is viewed as sampling at the intrinsic information rate as opposed to the more redundant Nyquist rate. The tractable recovery of \( K \)-sparse signal, \( \mathbf{x}(\omega, k) \) from the measurements, \( \mathbf{y}(\omega, k) \) requires the sensing matrix, \( \Phi(\omega) \) to obey the restricted isometry property (RIP) [7]. For the reconstruction, the sensing matrix, \( \Phi(\omega) \) must satisfy RIP of order \( K \) for all \( K \)-sparse signal, \( \mathbf{x}(\omega, k) \), if there exists a constant, \( \delta_K \in (0, 1) \) such that

\[
(1 - \delta_K) \| x(\omega, k) \|_2^2 \leq \| \Phi(\omega)x(\omega, k) \|_2^2 \leq (1 + \delta_K) \| x(\omega, k) \|_2^2
\]

where \( \| \cdot \|_2 \) denotes \( \ell_2 \) norm [7]. For notational convenience, the index \( \omega \) and \( k \) will be dropped when discussing the time-frequency representation of the signals.
The CS recovery is performed to obtain the sparse solution \( \hat{x} \) from \( y \). For a fixed value of \( \lambda \), the denoising problem is formulated as an optimisation problem

\[
\begin{align*}
\{ \hat{x} (\lambda) &= \arg \min_{x \in \mathbb{R}^N} f(x, \lambda) \\
\text{where } f(x, \lambda) &= ||y - \Phi x||_2^2 + \lambda ||x||_1
\end{align*}
\] (6)

The regularization parameter \( \lambda \) controls the sparsity of the reconstructed (denoised) signal \( \hat{x} \), which in turn regulates how well the signal is denoised. For instance, a large value of \( \lambda \) makes the output more sparse but potentially at the expense of speech distortion [16]. An optimal choice of \( \lambda \) allows the trade-off between smoothness of the reconstructed signal and similarity to the original signal akin to balancing noise suppression and signal distortion. In this paper, we employ the accelerated proximal gradient method as the first optimization level to solve the problem (6). Then the set of sparse solutions are optimized with objective measure such as the PESQ as the cost function.

3. Proximal gradient method and accelerated proximal gradient method with updated step size. For a fixed value of \( \lambda \), the cost function in (6) can be expressed as

\[
f(x, \lambda) = g(x) + h(x, \lambda)
\] (7)

which includes two parts. The first part \( g(x) = ||y - \Phi x||_2^2 \) is a differentiable and convex function with respect to \( x \) while the second part \( h(x, \lambda) = \lambda ||x||_1 \) is non-differentiable. The problem (6) has previously been solved using the interior point method [16], [12], which requires solving the Newton system that has high computational complexity.

Here, we propose to solve the problem by using the first order proximal gradient method with Barzilai and Borwein step size [4] for each iteration. The proximal gradient method is employed when the objective function has one component \( h(x, \lambda) \) which is convex but not differentiable. For each iteration \( k \), we find an updated vector that minimizes the cost function \( g(x) \) using the gradient method, based on the vector \( x^{(k-1)} \) at the \( k-1 \) iteration:

\[
v^{(k)} = x^{(k-1)} - \beta_{k-1} \nabla g(x^{(k-1)}).\] (8)

Next, the vector \( x^{(k)} \) is chosen as the one closest to \( v^{(k)} \) according to the Euclidean form which minimizes the objective function \( h(x, \lambda) \). This can be done by defining an approximal operator for the function \( h(x) \) as:

\[
x^{(k)} = \text{prox } h(v^{(k)}, \lambda) = \arg \min_{u} \left( h(u, \lambda) + \frac{1}{2} ||u - v^{(k)}||_2^2 \right)
\] (9)

For \( 1 \leq i \leq N \), the \( i \)th component of the vector \( x^{(k)} \) can be obtained by solving a one dimensional optimization problem:

\[
x_i^{(k)} = \arg \min_{u_i} \left( \lambda |u_i| + \frac{1}{2} (u_i - u_i^{(k)})^2 \right).
\] (10)
If $u_i \geq 0$, the problem (10) becomes:

$$x_i^{(k)} = \arg \min_{u_i} \left( \lambda u_i + \frac{1}{2} \left( u_i^2 - 2 u_i v_i^{(k)} + (v_i^{(k)})^2 \right) \right)$$

$$= \arg \min_{u_i} \left( \frac{1}{2} u_i^2 + (\lambda - v_i^{(k)}) u_i + (v_i^{(k)})^2 \right) = v_i^{(k)} - \lambda. \quad (11)$$

The solution for (11) for the case $u_i \geq 0$ becomes:

$$x_i^{(k)} = \begin{cases} v_i^{(k)} - \lambda, & \text{if } v_i^{(k)} - \lambda > 0 \text{ or } v_i^{(k)} > \lambda \\ 0, & \text{if } v_i^{(k)} \leq \lambda. \end{cases} \quad (12)$$

Similarly, the solution for (11) for the case $u_i \leq 0$ becomes:

$$x_i^{(k)} = \begin{cases} v_i^{(k)} + \lambda, & \text{if } v_i^{(k)} + \lambda < 0 \text{ or } v_i^{(k)} < -\lambda \\ 0, & \text{if } v_i^{(k)} \geq -\lambda. \end{cases} \quad (13)$$

As such, the solution for (9) can be expressed in the form:

$$x_i^{(k)} = \begin{cases} v_i^{(k)} - \lambda, & \text{if } v_i^{(k)} > \lambda \\ v_i^{(k)} + \lambda, & \text{if } v_i^{(k)} < -\lambda \\ 0, & \text{if } |v_i^{(k)}| \leq \lambda. \end{cases} \quad (14)$$

An equivalent vector formulation for $x^{(k)}$ based on coefficients in (14) can be given in the form

$$x^{(k)} = \text{sign}(v^{(k)}) \odot \max \left( 0, |v^{(k)}| - \lambda \right) \quad (15)$$

where “$\odot$” denotes the element-by-element multiplication between two vectors.

The proximal gradient method for optimising the problem (6) can be summarised as follow:

**Procedure 1**: Proximal gradient method for problem (6)

- Step 1: Initialize the iteration $k = 1$ the initial vector $x^{(0)}$ and a step size $\beta_0$.
- Step 2: Obtain the gradient vector $\nabla g(x^{(k-1)})$ for the function $g(x)$ as

$$\nabla g(x^{(k-1)}) = 2 \Phi^T (\Phi x^{(k-1)} - y). \quad (16)$$

- Step 3: Obtain the update vector $v^{(k)}$ that minimises the objective function $g(x)$ starting from $x^{(k-1)}$ using the gradient descent method as in (8),

$$v^{(k)} = x^{(k-1)} - \beta_{k-1} \nabla g(x^{(k-1)}) \quad (17)$$

where $\beta_k$ is the step size for the $k^{th}$ iteration.

- Step 4: Calculate the updated vector $x^{(k)}$ that is closest to $v^{(k)}$ and minimizes $h(x, \lambda)$ as in (14),

$$x^{(k)} = \text{sign}(v^{(k)}) \odot \max \left( 0, |v^{(k)}| - \lambda \right) \quad (18)$$

- Step 4: If

$$|f(x^{(k)}, \lambda) - f(x^{(k-1)}, \lambda)| < \epsilon, \quad (19)$$

where $\epsilon$ is a small tolerance, then stop the procedure and output the coefficient vector $x^{(k)}$. Otherwise, set $k + 1 \rightarrow k$, update the step size $\beta_k$ and return to Step 2. \qed
An important step of Procedure 1 is the step size strategy for $\beta_k$ in Step 3. Similar to the steepest descent algorithm, the procedure is guaranteed to converge for a constant step size $\beta_k = 1/L$ where $L$ is a Lipschitz constant. The convergence for Procedure 1 can be improved by updating the step size in each iteration [9], [8]. Here, we implement the Barzilai and Borwein (BB) method [4] to calculate the step size for each iteration. The method estimates the scalar approximation of the Hessian matrix and has low computational complexity,

$$\beta_k = \frac{||x^k - x^{k-1}||^2_2}{(x^k - x^{k-1})^T(\nabla g(x^k) - \nabla g(x^{k-1}))}. \quad (20)$$

The proximal gradient method exhibits the convergence rate $O(1/k)$. Here, we propose to use the accelerated [1] proximal gradient algorithm to further improve the convergence of the proximal gradient algorithm. In particular, the accelerated proximal gradient algorithm has approximately the same complexity per iteration as the proximal gradient algorithm, but with a much faster convergence rate resulting in lower total computational complexity for convergence. For each iteration of the accelerated proximal gradient algorithm, the new coefficient vector is obtained from the combination of the coefficient vector from Procedure 1 and the previous coefficient vector. The procedure for the accelerated proximal gradient algorithm can be summarized as follows:

**Procedure 2**: Accelerated proximal gradient method for problem (6)

- **Step 1**: Initialize the iteration $k = 1$ and the initial coefficient vector $z^{(0)}$.
- **Step 2**: Obtain the gradient vector $\nabla g(z^{(k-1)})$ for the function $g(z)$,

$$\nabla g(z^{(k-1)}) = 2\Phi^T(\Phi z^{(k-1)} - y). \quad (21)$$

- **Step 3**: Obtain the coefficient vector $x^{(k)}$ for the $k^{th}$ iteration starting from $z^{(k-1)}$ as in Steps 3 and 4 of Procedure 1:

$$\begin{cases} 
    v^{(k)} &= z^{(k-1)} - \beta_{k-1} \nabla g(z^{(k-1)}) \\
    x^{(k)} &= \text{sign}(v^{(k)}) \odot \max(0, |v^{(k)}| - \lambda)
\end{cases} \quad (22)$$

where $\beta_k$ is the step size.

- **Step 4**: The coefficient vector $z^{(k)}$ for the $k^{th}$ iteration can be obtained by combining the coefficient vectors $x^{(k-1)}$ and $x^{(k)}$ as

$$z^{(k)} = x^{(k)} + \frac{k - 1}{k + 2} (x^{(k)} - x^{(k-1)}). \quad (23)$$

- **Step 5**: Check for convergence, if

$$|f(z^{(k)}, \lambda) - f(z^{(k-1)}, \lambda)| < \epsilon, \quad (24)$$

where $\epsilon$ is a small tolerance, then stop the procedure and output the coefficient vector $z^{(k)}$. Otherwise, set $k + 1 \rightarrow k$ and return to Step 2. \Box

4. **Optimize $\lambda$ in (6) with respect to PESQ.** The coefficient vector in (6) depends on the weighting constant $\lambda$. As such, the problem becomes how to find the optimal value of $\lambda$ that optimizes the speech quality measures. In general, the assessment of speech quality can be classified in terms of subjective and objective evaluations. Subjective evaluation involves subjective listening test by some listeners while objective evaluation measures the numerical distance between the reference signal and the processed signal. One established method of evaluating the enhanced signal is using perceptual evaluation of speech quality (PESQ). PESQ is an automatic computation algorithm to replace human subjects in the evaluation
of the mean opinion score (MOS). The PESQ model considers how humans perceive
speech and has been widely used in the evaluation of speech quality. Note that
the proposed method does not restrict itself to PESQ but can be easily extended
to other objective measures.

Each optimal estimator is then evaluated against the affective measures, i.e.
PESQ performance. For a fixed value of $\lambda$, procedures 1 and 2 can be used to
solve (6). As such, we propose a two level optimization approach, which combines
an one dimensional optimization method to obtain an optimal $\lambda$ with the proximal
gradient method as developed in Procedures 1 and 2. The optimization problem
can be expressed as:

$$
\begin{align*}
\max_{\lambda,x(\lambda)} & \text{PESQ}(\lambda, x(\lambda)) \\
& x(\lambda) = \arg\min_{x \in \mathbb{R}^n} f(x, \lambda) \\
\text{where } & f(x, \lambda) = ||y - \Phi x||^2 + \lambda ||x||_1.
\end{align*}
$$

(25)

For simplicity, we denote the cost function as $\text{PESQ}(\lambda)$. Here, a line search
method that combines the golden search algorithm and quadratic interpolation is
employed to obtain the optimal solution for $\lambda$. The search for an optimum $\lambda$
can be summarized as follows.

**Procedure 3:** Search for an optimum $\lambda$ that maximizes the objective function
$\text{PESQ}(\lambda)$ or minimizes the objective function $-\text{PESQ}(\lambda)$ in an interval $[a, b]$.

- **Step 1:** Initialize a step size $s > 0$ and an accuracy level $\epsilon$.
- **Step 2:** Set $\lambda_0 = a$, $\lambda_1 = \lambda_0 + s$, $\lambda_2 = \lambda_0 + 2s$ and calculate $\text{PESQ}(\lambda_0)$,
$\text{PESQ}(\lambda_1)$ and $\text{PESQ}(\lambda_2)$. Search for $(\lambda_0, \lambda_1, \lambda_2)$ that $\text{PESQ}(\lambda_0) < \text{PESQ}(\lambda_1)$
and $\text{PESQ}(\lambda_1) > \text{PESQ}(\lambda_2)$. If there is $(\lambda_0, \lambda_1, \lambda_2)$ then there exits a local
maximum in the interval $[\lambda_0, \lambda_2]$ and go to step 3. Otherwise, set $\lambda_0 = \lambda_1$
and return to the beginning of Step 2. Continue to search until $\lambda_2$ reaches the
upper end $b$.
- **Step 3:** We have three points $\lambda_0$, $\lambda_1$ and $\lambda_2$ satisfying

$$\text{PESQ}(\lambda_0) < \text{PESQ}(\lambda_1), \text{PESQ}(\lambda_1) > \text{PESQ}(\lambda_2)$$

(26)

and $\lambda_0 < \lambda_1 < \lambda_2$. Thus, there exists a local maximum in the interval $[\lambda_0, \lambda_2]$. A
parabolic interpolation that goes through three points $(\lambda_0, \text{PESQ}(\lambda_0))$, $(\lambda_1, \text{PESQ}(\lambda_1))$ and $(\lambda_2, \text{PESQ}(\lambda_2))$ is then constructed to estimate for the
local maximum $\lambda$ in the interval $[\lambda_0, \lambda_2]$. The maximum of this parabola in
the interval $[\lambda_0, \lambda_2]$ can be obtained as

$$a = \lambda_1 - \frac{1}{2} \frac{\lambda_1 - \lambda_0)^2 \text{PESQ}(\lambda_1) - \text{PESQ}(\lambda_2) - (\lambda_1 - \lambda_2)^2 \text{PESQ}(\lambda_1) - \text{PESQ}(\lambda_0))}{(\lambda_1 - \lambda_0)(\lambda_1 - \lambda_2) (\text{PESQ}(\lambda_1) - \text{PESQ}(\lambda_0))}$$

(27)

with the corresponding cost function $\text{PESQ}(a)$. Next, progress to Step 5.
- **Step 5:** If $|\text{PESQ}(a) - \text{PESQ}(\lambda_1)| < \epsilon$, then set the local maximum as $\lambda_1$ and
go to Step 6. Otherwise, we have one of the following two cases
  - Case 1: The value of $a$ falls in the interval $[\lambda_0, \lambda_1]$. If $\text{PESQ}(a) < \text{PESQ}(\lambda_1)$
then set $\lambda_0 = a$ and $\text{PESQ}(\lambda_0) = \text{PESQ}(a)$. Otherwise, set $\lambda_2 = \lambda_1$, $\lambda_1 = a$, $\text{PESQ}(\lambda_2) = \text{PESQ}(\lambda_1)$ and $\text{PESQ}(\lambda_1) = \text{PESQ}(a)$.
  - Case 2: The value of $a$ falls in the interval $[\lambda_1, \lambda_2]$. If $\text{PESQ}(a) < \text{PESQ}(\lambda_1)$
then set $\lambda_2 = a$ and $\text{PESQ}(\lambda_2) = \text{PESQ}(a)$. Otherwise, set $\lambda_0 = \mu_1$, $\lambda_1 = a$, $\text{PESQ}(\lambda_0) = \text{PESQ}(\lambda_1)$ and $\text{PESQ}(\lambda_1) = \text{PESQ}(a)$.

For both cases, the new set of three points $(\lambda_0, \text{PESQ}(\lambda_0))$, $(\lambda_1, \text{PESQ}(\lambda_1))$
and $(\lambda_2, \text{PESQ}(\lambda_2))$ satisfies the constraints in (26). Thus, return to Step 4.
• Step 6: If the search in Step 3 reaches the upper end $b$ of $\lambda$, then set $\lambda_{opt}$ as the lowest of all the local minimum and stop the procedure. Otherwise, continue with the next If PESQ($a$) < PESQ($\lambda_1$) then let $\lambda_1 = a$. Set the optimum step size $\lambda^{(i)}$ as $\lambda^{(i)} = \lambda_1$ and stop the procedure. □

The advantage of Procedure 3 is that only the cost function PESQ($\lambda$) is required to be calculated. In addition, by employing parabolic interpolation, the optimum $\lambda_{opt}$ can be found with just a few calculations of the cost function.

5. Simulation results.

5.1. Experimental settings. In this section, extensive experiments were conducted to evaluate the performance of the proposed approach in different scenarios. Three different types of noise sources from the NOISEX database, namely, babble, destroyer and white noise were tested over a range of SNR from 0 dB to 20 dB, in steps of 5 dB. The noise types were chosen to represent the different degree of non-stationarity noise encountered in the real world. Five male and five female speech signals from the TIMIT database [2] were used in the simulation. The performance was evaluated by using the PESQ and STOI [21, 24]. As discussed, PESQ gives a score on the perceived speech quality, i.e., aspects of naturalness of the processed speech and STOI measures the overall intelligibility of speech. Naturally, an increase in both objective measures indicates the processed speech has a higher level of naturalness and intelligibility. In addition, the window length are chosen as 256 and 512 points with 50% oversampling and the compressive ratio $M/N$ is set to 0.9.

5.2. Convergence analysis. We first compare the performance of the three methods that are used to solve the optimization problem (6), namely the proximal gradient method as in Procedure 1, the accelerated proximal gradient method as in Procedure 2 and the interior point method [12]. The value of $\lambda$ is fixed at $\lambda = 0.95$.

Figures 1 and 2 show the convergence curves for the proximal gradient, the accelerated proximal gradient and the interior point methods for the babble noise with 0 dB. For comparison purpose, all the methods have the same starting point, with the initial vector $x = 0$. The window length $L$ in Figures 1 and 2 is chosen as 256 and 512, respectively. Note that the complexity per-iteration is different for the gradient methods and the interior point method with the complexity per interation for the interior point method is higher. Also, all the algorithms converge with the same accuracy. Notably, the convergence curves show that the accelerated proximal gradient method converges faster than the proximal gradient method and the interior point method, with the proximal gradient being the slowest to converge. As such, the accelerated proximal gradient method requires a lower number of iterations for convergence, which in turn leads to a lower computational complexity than the other two methods.

Figure 3 shows the convergence curves for the three methods with the destroyer noise and SNR of 0dB. The window length is chosen as $L = 512$. The results show the same consistency in the convergence performance with the accelerated proximal gradient method requiring a smaller number of iterations for convergence than the proximal gradient and the interior point methods. This naturally results in a lower computational complexity for the convergence of the proposed algorithm.
5.3. Computational complexity. Table 1 shows the computational time comparison for the accelerated proximal gradient, the proximal gradient and the interior point methods with the babble noise, destroyer noise and the white noise for different SNR and $L = 256$. The calculations are obtained by using Matlab on an Intel Core i7-7700 CPU. The computational complexity for the accelerated proximal gradient and the proximal gradient methods are significantly lower than the interior point method proposed in [12] as the proximal gradient approach requires only the calculation of the gradient vectors in Equations (16) and (21). As a comparison, the results show that the time required for the accelerated proximal gradient method is approximately three times faster than the interior point method. In addition, the time required for the accelerated proximal gradient method is lower than the proximal gradient method. It is interesting to point out that for the accelerated proximal method, slightly longer time is needed for poor SNR conditions compared to higher SNRs. For the interior point method, all scenarios require more than 6s for the solution to converge, irrespective of the SNRs.

Table 2 shows the computational time comparison for all the cases when the window length is double to $L = 512$. Similar to the previous case, the accelerated proximal gradient method requires a much lower time when compared to the proximal gradient method and the interior point method for convergence.

5.4. PESQ and STOI scores. We now compare the performance of PESQ and STOI for different value of $\lambda$. The optimization problem (25) is optimized using Procedure 3. Tables 3-5 tabulate the PESQ and STOI scores for the three types of noise, namely babble, destroyer and white noise for different SNR levels with the window length $L = 256$. For comparison purpose, the PESQ and STOI for commonly fixed values of $\lambda$, namely $\lambda = 0.8$ and $\lambda = 0.9$ were also presented. The results show improvement for both the PESQ and STOI when $\lambda$ is optimized compared to the non-optimized cases. An observation made was that the converged value of $\lambda$ was around $0.9 - 0.95$ for both babble noise and white noise across the SNRs. However, the converged values of $\lambda$ for destroyer noise were in the range of $0.89 - 0.95$. This suggests that the value of $\lambda$ should not be held constant but optimized according to the noise profile. In addition, the values of PESQ and STOI are significantly higher for the processed signals in comparison with the un-processed signal, resulting in an improvement in speech quality.

Tables 6-8 show the PESQ and STOI scores different noise types and SNR when the window length is increased to $L = 512$. Similar to the previous cases, the speech quality is improved by optimizing the value of $\lambda$ for all the three types of noise, namely babble, destroyer and white noise for different SNR. Also, the processed signals have much better PESQ and STOI values when compared to un-processed signals.

6. Conclusions. This paper proposes a two level optimization approach to incorporate objective quality measures in compressive speech enhancement. The proposed method combines the accelerated proximal gradient method and a one dimensional optimization approach to solve the sparse reconstruction and search for an optimal regularization parameter. By incorporating objective quality measures in the optimization process, the reconstructed output is not only sparse but also maintains the highest objective quality score possible for this type of method. Experimental results show that the proposed method has low computational complexity in
comparison with conventional approach of using the interior point method. In addition, the compressive speech enhancement consistently show score improvement in objectives measures in different noisy environments compared to the non-optimized method.

Acknowledgments. Siow Yong Low would like to acknowledge the support of the Fundamental Research Grant Scheme (FRGS/1/2019/TK04/USMC/02/1) under the auspices of the Malaysian Ministry of Higher Education (MOHE).

REFERENCES

[1] A. Beck and M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problem, *SIAM Journal on Imaging Sciences*, 2 (2009), 183–202.
[2] J. Benesty and Y. Huang, A Perspective on Single-Channel Frequency-Domain Speech Enhancement, San Rafael: Morgan and Claypool Publishers, 2010.
[3] S. F. Boll, Suppression of acoustic noise in speech using spectral subtraction, *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-27 (1979), 113–120.
[4] O. Burdakov, Y. Dai and N. Huang, Stabilized Barzilai-Borwein method, *J. Comp. Math.*, 37 (2019), 916–936.
[5] E. J. Candes, J. Romberg and T. Tao, Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information, *IEEE Transactions on Information Theory*, 52 (2006), 489–509.
[6] E. J. Candes and T. Tao, Near-optimal signal recovery from random projections: universal encoding strategies, *IEEE Transactions on Information Theory*, 52 (2006), 5406–5425.
[7] E. J. Candes and M. B. Wakin, An introduction to compressive sampling, *IEEE Signal Processing Magazine*, (2008), 21–30.
[8] H. H. Dam and A. Cantoni, Interior point method for optimum zero-forcing beamforming with per-antenna power constraints and optimal step size, *Signal Processing*, 106 (2015), 10–14.
[9] H. H. Dam and S. Nordholm, Accelerated gradient with optimal step size for second-order blind signal separation, *Multidimens. Syst. Signal Process.*, 29 (2018), 903–919.
[10] T. Esch and P. Vary, Efficient musical noise suppression for speech enhancement system, 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, (2009), 4409–4412.
[11] P. K. Ghosh, A. Tsiartas and S. Narayanan, Robust voice activity detection using long-term signal variability, *IEEE Transactions on Audio, Speech and Language Processing*, 19 (2011), 600–613.
[12] S. J. Kim, K. Koh, M. Lustig, S. Boyd and D. Gorinevsky, An interior-point method for large-scale $l_1$-regularized least squares, *IEEE Journal of Selected Topics in Signal Processing*, 1 (2007), 606–617.
[13] H. Li, C. Fang and Z. Lin, Accelerated first-order optimization algorithms for machine learning, *Proceedings of the IEEE*, (2020), 1–16.
[14] P. C. Loizou, *Speech Enhancement: Theory and Practice*, CRC press, Boca Raton, 2013.
[15] S. Y. Low, Compressive speech enhancement in the modulation domain, *Speech Communication*, 102 (2018), 87–99.
[16] S. Y. Low, D. S. Pham and S. Venkatesh, Compressive speech enhancement, *Speech Communication*, 55 (2013), 757–768.
[17] R. Martin, Noise power spectral density estimation based on optimal smoothing and minimum statistics, *IEEE Transactions on Speech and Audio Processing*, 9 (2001), 504–512.
[18] R. Miyazaki, H. Saruwatari, T. Inoue, K. Shikano and K. Kondo, Musical-noise-free speech enhancement: Theory and evaluation, 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), (2012), 4565–4568.
[19] M. Nazih, K. Minaoui and P. Comon, Using the proximal gradient and the accelerated proximal gradient as a canonical polyadic tensor decomposition algorithms in difficult situations, *Signal Processing*, 171 (2020), 107472.
[20] N. Parikh and S. Boyd, Proximal Algorithms, *Foundation and Trends in Optimization*, 1 (2013), 123–231.
[21] A. W. Rix, J. G. Beerends, M. P. Hollier and A. P. Hekstra, Perceptual evaluation of speech quality (PESQ) - a new method for speech quality assessment of telephone networks and codecs, 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing, Proceedings (Cat. No.01CH37221), 2 (2001), 749–752.
[22] M. Schmidt, Least squares optimization with 1-norm regularization, Technical Report CSP542B, 2005.
[23] Y. Shi, S. Y. Low and K. F. C. Yiu, Hyper-parameterization of sparse reconstruction for speech enhancement, Applied Acoustics, 138 (2018), 72–79.
[24] C. H. Taal, R. C. Hendriks, R. Heusdens and J. Jensen, A short-time objective intelligibility measure for time-frequency weighted noisy speech, 2010 IEEE International Conference on Acoustics, Speech and Signal Processing, (2010), 4214–4217.
[25] R. Tibshirani, Regression shrinkage and selection via the lasso, J. Roy. Statist. Soc. Ser. B, 58 (1996), 267–288.
[26] M. Torcoli, An improved measure of musical noise based on spectral kurtosis, 019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), (2019), 90–94.
[27] D. Wu, W. Zhu and M. N. S. Swamy, A compressive sensing method for noise reduction of speech and audio signals, 2011 IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS), (2011), 1–4.
[28] Z. Zhang, Y. Xu, J. Yang, X. Li and D. Zhang, A Survey of Sparse Representation: Algorithms and Applications, IEEE Access, 3 (2015), 490–530.

Received November 2020; revised May 2021; early access August 2021.

E-mail address: H.dam@curtin.edu.au
E-mail address: sy.low@soton.ac.uk
E-mail address: s.nordholm@curtin.edu.au

![Figure 1. Convergence for accelerated proximal gradient, proximal gradient methods and interior point methods for babble noise with 0 dB and $L = 256$.](image)
**Figure 2.** Convergence for the proximal gradient, the accelerated proximal gradient, and the interior point methods for babble noise with 0 dB and $L = 512$.

**Figure 3.** Convergence for the proximal gradient, the accelerated proximal gradient, and the interior point methods for destroyer noise with 0 dB and $L = 512$. 
ACCELERATED PROXIMAL GRADIENT FOR SPARSE SPEECH RECONSTRUCTION

Table 1. Complexity comparison between the proximal gradient, the accelerated proximal gradient, and the interior point methods for babble noise, destroyer noise and white noise with window length $L = 256$.

| Noise type       | SNR | Accelerated Proximal Gradient | Proximal Gradient | Interior Point Method |
|------------------|-----|-------------------------------|-------------------|-----------------------|
| Babble noise     | 0dB | 3.1307s                       | 3.3139s           | 7.2640s               |
|                  | 5dB | 2.7209s                       | 2.8722s           | 7.0281s               |
|                  | 10dB| 2.3887s                       | 2.4476s           | 6.8677s               |
|                  | 15dB| 2.2449s                       | 2.4057s           | 6.8233s               |
|                  | 20dB| 2.0481s                       | 2.1050s           | 6.6695s               |
| Destroyer noise  | 0dB | 2.8322s                       | 2.9187s           | 6.9363s               |
|                  | 5dB | 2.4799s                       | 2.5386s           | 6.8301s               |
|                  | 10dB| 2.2675s                       | 2.4413s           | 6.7417s               |
|                  | 15dB| 2.1390s                       | 2.2119s           | 6.7070s               |
|                  | 20dB| 1.8859s                       | 1.9688s           | 6.4216s               |
| White noise      | 0dB | 3.4491s                       | 3.5234s           | 6.6548s               |
|                  | 5dB | 2.8229s                       | 2.9723s           | 6.9340s               |
|                  | 10dB| 2.5765s                       | 2.6288s           | 7.2333s               |
|                  | 15dB| 2.3393s                       | 2.4726s           | 7.0217s               |
|                  | 20dB| 1.9912s                       | 2.0732s           | 6.5130s               |

Table 2. Complexity comparison between the accelerated proximal gradient, the proximal gradient and the interior point methods for babble noise, destroyer noise and white noise with window length $L = 512$.

| Noise type       | SNR | Accelerated Proximal Gradient | Proximal Gradient | Interior Point Method |
|------------------|-----|-------------------------------|-------------------|-----------------------|
| Babble noise     | 0dB | 0.8681s                       | 0.9342s           | 12.7778s              |
|                  | 5dB | 0.7779s                       | 0.8346s           | 12.5931s              |
|                  | 10dB| 0.7119s                       | 0.7730s           | 12.2826s              |
|                  | 15dB| 0.6637s                       | 0.7199s           | 12.0663s              |
|                  | 20dB| 0.6138s                       | 0.6703s           | 11.7910s              |
| Destroyer noise  | 0dB | 0.8096s                       | 0.8760s           | 12.4143s              |
|                  | 5dB | 0.7330s                       | 0.7863s           | 12.4028s              |
|                  | 10dB| 0.6709s                       | 0.7329s           | 12.0540s              |
|                  | 15dB| 0.6263s                       | 0.6908s           | 11.9282s              |
|                  | 20dB| 0.5950s                       | 0.6550            | 11.8206s              |
| White noise      | 0dB | 0.9592s                       | 1.0401s           | 11.9704s              |
|                  | 5dB | 0.8137s                       | 0.8761s           | 12.5119s              |
|                  | 10dB| 0.7049s                       | 0.7656s           | 12.8533s              |
|                  | 15dB| 0.6503s                       | 0.7136s           | 12.3004s              |
|                  | 20dB| 0.6193s                       | 0.6818s           | 11.9545s              |
Table 3. PESQ and STOI performance for different SNR with babble noise and $L = 256$.

| SNR | Methods       | $\lambda$ | PESQ  | STOI  |
|-----|---------------|-----------|-------|-------|
| 0 dB| Optimized     | $\lambda = 0.9549$ | 2.0328 | 0.7147 |
|     | Fixed value   | $\lambda = 0.8$  | 2.0073 | 0.7032 |
|     | Fixed value   | $\lambda = 0.9$  | 2.0241 | 0.7103 |
|     | Unprocessed   | --         | 1.8938 | 0.7145 |
| 5 dB| Optimized     | $\lambda = 0.9449$ | 2.4100 | 0.8200 |
|     | Fixed value   | $\lambda = 0.8$  | 2.3896 | 0.8107 |
|     | Fixed value   | $\lambda = 0.9$  | 2.3996 | 0.8170 |
|     | Unprocessed   | --         | 2.2203 | 0.8130 |
| 10 dB| Optimized    | $\lambda = 0.9549$ | 2.7702 | 0.8990 |
|      | Fixed value   | $\lambda = 0.8$  | 2.7522 | 0.8918 |
|      | Fixed value   | $\lambda = 0.9$  | 2.7639 | 0.8974 |
|      | Unprocessed   | --         | 2.5434 | 0.8899 |
| 15 dB| Optimized    | $\lambda = 0.9525$ | 3.1247 | 0.9504 |
|      | Fixed value   | $\lambda = 0.8$  | 3.0937 | 0.9455 |
|      | Fixed value   | $\lambda = 0.9$  | 3.1144 | 0.9489 |
|      | Unprocessed   | --         | 2.8556 | 0.9423 |
| 20 dB| Optimized    | $\lambda = 0.9549$ | 3.4425 | 0.9767 |
|      | Fixed value   | $\lambda = 0.8$  | 3.3898 | 0.9731 |
|      | Fixed value   | $\lambda = 0.9$  | 3.4317 | 0.9757 |
|      | Unprocessed   | --         | 3.1674 | 0.9734 |
Table 4. PESQ and STOI performance for different SNR with destroyer noise and $L = 256$.

| SNR  | Method   | $\lambda$ | PESQ  | STOI  |
|------|----------|-----------|-------|-------|
| 0 dB | Optimized | $0.8949$ | 2.1629 | 0.7532 |
|      | Fixed    | 0.8       | 2.1543 | 0.7448 |
|      | Fixed    | 0.9       | 2.1456 | 0.7497 |
|      | Unprocessed | —       | 1.9271 | 0.7524 |
| 5 dB | Optimized | $0.8951$ | 2.5370 | 0.8373 |
|      | Fixed    | 0.8       | 2.5186 | 0.8267 |
|      | Fixed    | 0.9       | 2.5283 | 0.8325 |
|      | Unprocessed | —       | 2.2955 | 0.8281 |
| 10 dB| Optimized | $0.8749$ | 2.8704 | 0.9001 |
|      | Fixed    | 0.8       | 2.8543 | 0.8933 |
|      | Fixed    | 0.9       | 2.8677 | 0.8985 |
|      | Unprocessed | —       | 2.6132 | 0.8902 |
| 15 dB| Optimized | $0.9451$ | 3.1914 | 0.9468 |
|      | Fixed    | 0.8       | 3.1611 | 0.9412 |
|      | Fixed    | 0.9       | 3.1876 | 0.9455 |
|      | Unprocessed | —       | 2.9256 | 0.9382 |
| 20 dB| Optimized | $0.9451$ | 3.4868 | 0.9737 |
|      | Fixed    | 0.8       | 3.4427 | 0.9696 |
|      | Fixed    | 0.9       | 3.4722 | 0.9726 |
|      | Unprocessed | —       | 3.2468 | 0.9697 |

Table 5. PESQ and STOI performance for different SNR with white noise and $L = 256$.

| SNR  | Method   | $\lambda$ | PESQ  | STOI  |
|------|----------|-----------|-------|-------|
| 0 dB | Optimized | $0.9331$ | 2.0119 | 0.7661 |
|      | Fixed    | 0.8       | 1.9895 | 0.7519 |
|      | Fixed    | 0.9       | 2.0042 | 0.7619 |
|      | Unprocessed | —       | 1.6665 | 0.7377 |
| 5 dB | Optimized | $0.9451$ | 2.3972 | 0.8615 |
|      | Fixed    | 0.8       | 2.3716 | 0.8492 |
|      | Fixed    | 0.9       | 2.3913 | 0.8580 |
|      | Unprocessed | —       | 1.9615 | 0.8387 |
| 10 dB| Optimized | $0.9451$ | 2.8102 | 0.9275 |
|      | Fixed    | 0.8       | 2.7735 | 0.9183 |
|      | Fixed    | 0.9       | 2.7976 | 0.9246 |
|      | Unprocessed | —       | 2.2989 | 0.9146 |
| 15 dB| Optimized | $0.9349$ | 3.1973 | 0.9652 |
|      | Fixed    | 0.8       | 3.1472 | 0.9594 |
|      | Fixed    | 0.9       | 3.1844 | 0.9636 |
|      | Unprocessed | —       | 2.6442 | 0.9613 |
| 20 dB| Optimized | $0.9501$ | 3.5007 | 0.9858 |
|      | Fixed    | 0.8       | 3.4286 | 0.9797 |
|      | Fixed    | 0.9       | 3.4796 | 0.9826 |
|      | Unprocessed | —       | 2.9839 | 0.9845 |
Table 6. PESQ and STOI performance for different SNR with babble noise and $L = 512$.

| SNR | Methods          | $\lambda$ | PESQ  | STOI  |
|-----|------------------|-----------|-------|-------|
| 0 dB| Optimized        | 0.9601    | 2.0699| 0.7234|
|     | Fixed value      | 0.8       | 2.0525| 0.7129|
|     | Fixed value      | 0.9       | 2.0634| 0.7212|
|     | Unprocessed      | --        | 1.8938| 0.7145|
| 5 dB| Optimized        | 0.9601    | 2.4185| 0.8282|
|     | Fixed value      | 0.8       | 2.4084| 0.8195|
|     | Fixed value      | 0.9       | 2.4150| 0.8258|
|     | Unprocessed      | --        | 2.2203| 0.8130|
| 10 dB| Optimized       | 0.9079    | 2.7672| 0.9064|
|      | Fixed value      | 0.8       | 2.7529| 0.8996|
|      | Fixed value      | 0.9       | 2.7586| 0.9045|
|      | Unprocessed      | --        | 2.5434| 0.8899|
| 15 dB| Optimized       | 0.9077    | 3.1187| 0.9540|
|      | Fixed value      | 0.8       | 3.0736| 0.9507|
|      | Fixed value      | 0.9       | 3.0790| 0.9530|
|      | Unprocessed      | --        | 2.8556| 0.9423|
| 20 dB| Optimized       | 0.9601    | 3.3898| 0.9785|
|      | Fixed value      | 0.8       | 3.3703| 0.9760|
|      | Fixed value      | 0.9       | 3.3822| 0.9775|
|      | Unprocessed      | --        | 3.1674| 0.9734|
Table 7. PESQ and STOI performance for different SNR with destroyer noise and 512 subbands.

| SNR | Method s | λ  | PESQ | STOI |
|-----|----------|----|------|------|
| 0 dB | Optimized | 0.7601 | 2.2328 | 0.7629 |
|     | Fixed value | 0.8 | 2.2256 | 0.7602 |
|     | Fixed value | 0.9 | 2.2078 | 0.7622 |
|     | Unprocessed | -- | 1.9271 | 0.7524 |
| 5 dB | Optimized | 0.7700 | 2.5651 | 0.8441 |
|     | Fixed value | 0.8 | 2.5589 | 0.8414 |
|     | Fixed value | 0.9 | 2.5599 | 0.8436 |
|     | Unprocessed | -- | 2.2955 | 0.8281 |
| 10 dB | Optimized | 0.7700 | 2.8773 | 0.9084 |
|      | Fixed value | 0.8 | 2.8699 | 0.9056 |
|      | Fixed value | 0.9 | 2.8742 | 0.9081 |
|      | Unprocessed | -- | 2.6132 | 0.8902 |
| 15 dB | Optimized | 0.8301 | 3.1775 | 0.9530 |
|      | Fixed value | 0.8 | 3.1710 | 0.9509 |
|      | Fixed value | 0.9 | 3.1732 | 0.9529 |
|      | Unprocessed | -- | 2.9256 | 0.9382 |
| 20 dB | Optimized | 0.9270 | 3.4819 | 0.9768 |
|      | Fixed value | 0.8 | 3.4375 | 0.9750 |
|      | Fixed value | 0.9 | 3.4469 | 0.9766 |
|      | Unprocessed | -- | 3.2468 | 0.9697 |

Table 8. PESQ and STOI performance for different SNR with white noise and 512 subbands.

| SNR | Methods | λ  | PESQ | STOI |
|-----|---------|----|------|------|
| 0 dB | Optimized | 0.8599 | 2.0454 | 0.7829 |
|     | Fixed value | 0.8 | 2.0395 | 0.7721 |
|     | Fixed value | 0.9 | 2.0403 | 0.7775 |
|     | Unprocessed | -- | 1.6665 | 0.7377 |
| 5 dB | Optimized | 0.8801 | 2.4148 | 0.8735 |
|     | Fixed value | 0.8 | 2.4129 | 0.8638 |
|     | Fixed value | 0.9 | 2.4101 | 0.8695 |
|     | Unprocessed | -- | 1.9615 | 0.8387 |
| 10 dB | Optimized | 0.9496 | 2.8009 | 0.9338 |
|      | Fixed value | 0.8 | 2.7937 | 0.9261 |
|      | Fixed value | 0.9 | 2.7948 | 0.9308 |
|      | Unprocessed | -- | 2.2989 | 0.9146 |
| 15 dB | Optimized | 0.9550 | 3.1967 | 0.9673 |
|      | Fixed value | 0.8 | 3.1421 | 0.9623 |
|      | Fixed value | 0.9 | 3.1514 | 0.9652 |
|      | Unprocessed | -- | 2.6442 | 0.9613 |
| 20 dB | Optimized | 0.9601 | 3.4393 | 0.9864 |
|      | Fixed value | 0.8 | 3.3930 | 0.9807 |
|      | Fixed value | 0.9 | 3.4175 | 0.9823 |
|      | Unprocessed | -- | 2.9839 | 0.9845 |