A Penalty Function Promoting Sparsity Within and Across Groups

İlker Bayram and Savaşkan Bulek

***Abstract*—We introduce a new penalty function that promotes signals composed of a small number of active groups, where within each group, only a few high magnitude coefficients are non-zero. We derive the threshold function associated with the proposed penalty and study its properties. We discuss how the proposed penalty/threshold function can be useful for signals with isolated non-zeros, such as audio with isolated harmonics along the frequency axis, or reflection functions in exploration seismology where the non-zeros occur on the boundaries of subsoil layers. We demonstrate the use of the proposed penalty/threshold functions in a convex denoising and a non-convex deconvolution formulation. We provide convergent algorithms for both formulations and compare the performance with state-of-the-art methods.**

I. Introduction

Constraints or prior information derived from sparsity is widely used for regularization in signal processing. Depending on the application domain, the signal of interest may exhibit additional features than mere sparsity. In this paper, we consider signals whose coefficients can be clustered in a few groups where each group itself has few active members. Such a characteristic implies some dependence within groups. This is in contrast to plain sparsity, where the coefficients are treated independently. Sparse signals with isolated non-zeros form a class of signals that fall in the category of interest in this paper. We propose a prior function that promotes such signals and demonstrate how to use the function in basic inverse problems of potential interest.

Many natural phenomena can be associated with a sparse underlying process with isolated non-zero components. For instance, the DFT coefficients of a periodic signal are equidistant with respect to the frequency variable. Consequently, quasi-periodic audio signals like speech, music can be represented in the time-frequency domain (via linear transforms [3]) using components that appear isolated along the frequency axis (i.e., harmonics). Another example is related to reflection seismology, where one aims to discover the subsoil layers by sending seismic waves and processing the returning seismic trace [30]. The seismic trace can be modelled as the convolution of the input seismic wave and the reflection function. The reflection function is a sparse signal with non-zeros occuring due to difference in acoustic impedance between the boundaries of different layers. Since the layers are expected to have some non-zero thickness, the non-zeros, which occur at the boundaries, are isolated. Other than these natural signals, isolated sparsity is also relevant for designed systems. For instance, in frequency hopping systems, the parameters of the signal components are constant in between the hopping instances, during which transmission occurs [1]. Since transmission has to last for some finite amount of time, the hopping instances may be regarded as isolated non-zeros of a sparse signal.

In order to isolate the non-zeros, we work with non-overlapping groups of variables and process the groups independently. We propose a penalty whose threshold function (to be specified below) has the following properties :

- If the magnitudes of all variables in a group fall below a threshold, the whole group is set to zero.
- Otherwise, a group-dependent threshold is applied so as to eliminate the relatively insignificant coefficients in the group.

The group-dependent threshold serves to separate the large magnitude coefficients from the rest. Specifically, if there are k large-magnitude coefficients in the group, they are kept with little modification, while the rest are set to zero. For $k = 1$, this isolates the non-zeros within the group. We remark that this behavior is achieved with a non-adaptive penalty function and without reweighting.

To be more precise, assuming that the size of the groups is n, and that $x^{(i)} \in \mathbb{C}^n$ denotes the coefficients belonging to the i^{th} group of x, we define a penalty function for $\gamma \geq 0$ as,

$$\mathbf{P}_\gamma(x) = \sum_i P_\gamma\big(x^{(i)}\big), \tag{1}$$

where for $u \in \mathbb{C}^n$, P_γ is defined as,

$$P_\gamma(u) = \gamma \left(\sum_{i=1}^{n-1} \sum_{m=i+1}^{n} |u_i \, u_m| \right) + \|u\|_1. \tag{2}$$

The term enclosed in parentheses in (2) grows rapidly as the number of large magnitude coefficients in the group increases. Therefore P_γ strongly penalizes groups containing many large coefficients. Given this penalty, we describe how to realize the associated threshold function (or the proximity operator [10]) defined for $\lambda \geq 0$ as

$$\mathbf{T}_{\lambda,\gamma}(z) = \arg\min_x \ \frac{1}{2}\|z - x\|_2^2 + \lambda \, \mathbf{P}_\gamma(x). \tag{3}$$

We show that $\mathbf{T}_{\lambda,\gamma}$ is well-defined when $\lambda\,\gamma < 1$ and study its behavior. We also show that, as $\gamma \to 1/\lambda$, the threshold function suppresses all but the largest coefficient in each group, provided the magnitude of the largest coefficient exceeds the threshold λ. We demonstrate the use of the proposed penalty and the threshold function in a convex formulation for audio denoising and a non-convex formulation for non-blind deconvolution. We provide convergent algorithms for

İ. Bayram is with the Dept. of Electronics and Communications Eng., Istanbul Technical University, Istanbul, Turkey. E-mail : ibayram@itu.edu.tr. S. Bulek is with Qualcomm Atheros, Inc., Auburn Hills, MI, USA. E-mail : sbulek@gmail.com.

both formulations and demonstrate that the reconstructions perform favorably compared to those obtained using other penalties/threshold functions.

Related Work

The proposed penalty function may be regarded as a member of the family of group-based penalty functions (see e.g. [34, 18, 17, 5, 8, 29] for a sample of the literature). In contrast to our interest, many of these works seek to set whole groups of coefficients to zero, thus achieving sparsity across groups, and do not enforce sparsity within groups. For instance, the $\ell_{2,1}$ norm [18] is obtained by replacing $P_\gamma(x^{(i)})$ with $\|x^{(i)}\|_2$ in (1). The proximity operator associated with the $\ell_{2,1}$ norm sets a whole group to zero if the energy of the group is below a threshold but keeps the group with little modification otherwise. On the other hand, in the Elitist-Lasso (E-Lasso) formulation [18, 20] (see also [35] where the method is referred to as Exclusive-Lasso), the target signal contains few non-zeros within each group and sparsity is not enforced across groups. Sparsity within groups is also addressed by the sparse-group lasso (SGL) proposed in [29]. SGL uses a convex combination of an ℓ_1 norm and an $\ell_{2,1}$ norm as the penalty – it may also be interpreted as a sum of elastic-net-like penalties [36] applied to each group. Therefore SGL uses a convex penalty function. SGL was extended to non-overlapping groups and its performance is thoroughly analyzed in [24].

Another class of related penalties are based on correlations extracted from the observation matrix [31, 13]. Given an observation model of the form $y \approx H\,x$, the idea is to derive a positive semi-definite weighting matrix W from the correlations between the columns of H and use it to define a penalty of the form $x^T\,W\,x$. Since W does not depend on x, the penalty in [31, 13] is convex. The targeted effect is a uniform treatment of the components of x that produce similar responses. This contrasts with the proposed penalty because if two components of x have similar responses, the proposed penalty P_γ would prefer to single out one of the components and suppress the other. Another recent paper that takes into account correlations between the columns of H is [27]. A bivariate non-convex penalty is proposed so as to enforce sparsity stronger than alternative convex penalties, while maintaining the convexity of the overall problem. Sparsity within groups is not specifically sought in [27].

The penalty proposed in this paper, P_γ, is non-convex. However, its degree of non-convexity is controlled by the parameter γ and this in turn allows to formulate convex problems. As will be clarified in the sequel (see the proof of Prop. 1), P_γ can be related to the E-Lasso penalty. However, the E-Lasso penalty is convex and can be shown to contain an additive energy term, which in turn penalizes higher coefficients more strongly. Further, the E-Lasso threshold never sets the whole group to zero, unless the group is zero to start with (see [18], Remark-6). Thus, if a group consists entirely of noise, it will not be totally eliminated, even if it has components with small magnitudes. The threshold function associated with the proposed penalty function contains a deadzone such that if the coefficients in the group fall in the deadzone, then the whole group is eliminated. Therefore, the proposed penalty/threshold functions aim to achieve sparsity within and across groups.

Notation and Preliminaries

Throughout the paper, vectors are denoted using small case letters, as in x. The i^{th} component of x is denoted as x_i. We are interested in partitions of x into groups in this paper. We already used $x^{(i)}$ to denote the i^{th} subgroup of x. That is, for a length-4 vector $x = \big(x_1, \ldots, x_4\big)$, if we form two groups of size two, by collecting together consecutive components, we have $x^{(1)} = \big(x_1, x_2\big)$ and $x^{(2)} = \big(x_3, x_4\big)$. However, with the exception of Sec. II-E, the functions under study are separable with respect to groups. Therefore, whenever separability applies, we suppress the group superscript in $x^{(i)}$ and use x to simplify notation, with the understanding that the same discussion applies to all of the groups.

For a scalar $x \in \mathbb{C}$, the soft threshold function with threshold $\tau > 0$ is defined as,

$$\text{soft}(x, \tau) = \big(|x| - \tau\big)_+ \; \frac{x}{|x|}, \tag{4}$$

where, for $u \in \mathbb{R}$, we define,

$$(u)_+ = \max(u, 0). \tag{5}$$

If x is a vector, the soft thresholding operator applies to each component of x separately.

The proximity operator of a convex, lower semi-continuous function f is defined as [4, 10]

$$J_{\alpha f}(z) = \arg\min_x \frac{1}{2}\|z - x\|_2^2 + \alpha f(x). \tag{6}$$

We also refer to J as the threshold function, if f under discussion is a penalty function.

Throughout the paper, for a given length-n vector z (complex or real valued), we define the cost function $C_{\lambda,\gamma}(x|z)$ as

$$C_{\lambda,\gamma}(x|z) = \frac{1}{2}\|z - x\|_2^2 + \lambda\, P_\gamma(x), \tag{7}$$

where P_γ is given in (2). The threshold function $T_{\lambda,\gamma}(z)$ is defined, in line with (6), as,

$$T_{\lambda,\gamma}(z) = \arg\min_x C_{\lambda,\gamma}(x|z). \tag{8}$$

We remark that the penalty function used in this paper is not convex, but weakly convex. That is, the sum of the penalty function and a quadratic function is convex (see Defn. 1 in Sec. II-B). In order for the threshold function to be well-defined, the minimizer of $C_{\lambda,\gamma}(\cdot|z)$ (i.e., the point that minimizes $C_{\lambda,\gamma}(\cdot|z)$) must be unique. To ensure uniqueness, we will check that $C_{\lambda,\gamma}(\cdot|z)$ is strictly convex.

Outline

We motivate the proposed penalty function and derive the associated threshold function in Section II. We discuss how the non-convex penalty function may be employed to formulate a convex denoising problem with a sparsifying frame and present a minimization algorithm in Section III. In Section IV,

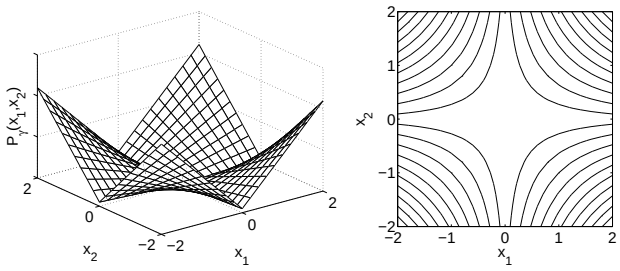

Fig. 1. Mesh and contour plots of the proposed $P_\gamma(x_1, x_2)$ for $\gamma = 10$. Notice that the function is not convex.

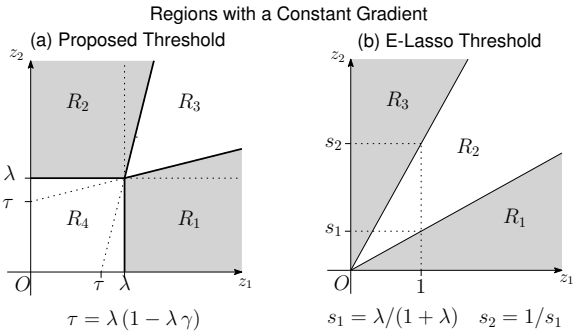

Fig. 2. The threshold functions associated with the proposed penalty and E-Lasso have constant gradients in the regions indicated above.

we present a non-convex deconvolution formulation, study the convergence of an iterative thresholding algorithm for the presented formulation and demonstrate its performance. Section V is the conclusion.

The webpage "http://web.itu.edu.tr/ibayram/SWAG/" contains the Matlab code used in the experiments and some additional material. The same Matlab code is also available at "https://doi.org/10.5281/zenodo.290588".

II. A Weakly Convex Penalty

The penalty function $\mathbf{P}_\gamma$ introduced in (1) is separable with respect to the groups and the groups are non-overlapping. Thanks to these properties, it suffices to study the component function $P_\gamma(x)$ and the associated threshold function $T_{\lambda,\gamma}$, with domain $\mathbb{R}^n$ or $\mathbb{C}^n$. Once $T_{\lambda,\gamma}$ is specified, $\mathbf{T}_{\lambda,\gamma}$ can be realized by applying $T_{\lambda,\gamma}$ to each group separately.

We start our discussion in Section II-A with penalty/threshold functions defined on $\mathbb{R}^2$, since this case is easier to visualize and interpret. After that, we generalize the discussion to $\mathbb{R}^n$ in Section II-B. A discussion of how to tune the parameters and a numerical demonstration of the discussions is provided in Sec. II-C. Extension of the study to $\mathbb{C}^n$ is done in Sec. II-D. Finally, in Sec. II-E, we briefly consider how the proposed penalty can be combined with $\ell_{2,1}$ norms to achieve a modified effect.

A. The Penalty and the Threshold Function on $\mathbb{R}^2$

1) The Penalty Function: For a fixed energy vector $x = (x_1, x_2)$, we seek a penalty function P such that,

- if $|x_1| \ll |x_2|$ or $|x_2| \ll |x_1|$, $P(x)$ assumes a low value,
- if $|x_1| \approx |x_2|$, $P(x)$ assumes a high value.

Observe that $|x_1\, x_2|$ satisfies these requirements. However, the function $|x_1\, x_2|$ is exactly zero when one of the components is zero. In order to penalize small non-zero components, we add an ℓ_1 term and propose the penalty

$$P_\gamma(x) = \gamma\, |x_1\, x_2| + \|x\|_1, \tag{9}$$

where $\gamma \geq 0$ is a tuning parameter. Notice that this is the restriction of the function in (2) to $\mathbb{R}^2$. Mesh and contour plots of this function for $\gamma = 10$ are shown in Fig. 1.

P_γ is not convex but it becomes convex when we add a quadratic term. Such functions are called weakly convex [32]. For $\lambda \leq 1/\gamma$, it can be shown that the function '$\|x\|_2^2/2 + \lambda\, P_\gamma(x)$' is convex (see Sec. II-B). Therefore, γ may be regarded as a parameter that controls how much P_γ deviates from being convex. As a consequence of the weak-convexity of P_γ, we find that if $\lambda < 1/\gamma$, then for a given $z \in \mathbb{R}^2$, the cost function $C_{\lambda,\gamma}(x|z)$ is strictly convex with respect to x. Thus, $T_{\lambda,\gamma}(z)$ is well-defined when $\lambda\, \gamma < 1$.

Before we further discuss the threshold function, we would like to compare P_γ to the Elitist-Lasso (E-Lasso) penalty function, which is known to favor large components in a group [18, 19, 20]. For groups of size two, the E-Lasso penalty is $P_{\text{EL}}(x) = \|x\|_1^2$. Expanding this, we can write,

$$P_{EL}(x) = 2|x_1\, x_2| + \|x\|_2^2. \tag{10}$$

Both P_{EL} and P_γ employ the term $|x_1\, x_2|$ and this term is responsible for the 'elitist' character of the penalties, by which we mean that the penalty assumes lower values for vectors with unequal components. The difference between the two penalties lies in the remaining components. P_γ contains an additive ℓ_1 term which helps enforce sparsity if the components have small magnitudes. In contrast, P_{EL} contains an additive energy term, which renders the overall penalty convex at the expense of penalizing high magnitude coefficients more strongly.

2) The Threshold Function: The threshold function $T_{\lambda,\gamma}$ can be derived via the optimality conditions for the minimization problem (8). Let us assume that $z_i \geq 0$, i.e., z lies in the first quadrant (extension to the other quadrants can be achieved by symmetry). Let $\hat{x} = T_{\lambda,\gamma}(z)$. For $\tau = \lambda\,(1 - \lambda\,\gamma)$, we can express $\hat{x} = (\hat{x}_1, \hat{x}_2)$ as follows.

$$\begin{aligned}
&\left.\begin{aligned}\hat{x}_1 &= z_1 - \lambda\\ \hat{x}_2 &= 0\end{aligned}\right\},\ \text{if}\ \left\{\begin{aligned}&z_1 \geq \lambda,\\ &z_2 \leq \lambda\,\gamma\, z_1 + \tau,\end{aligned}\right.\\
&\left.\begin{aligned}\hat{x}_1 &= 0\\ \hat{x}_2 &= z_2 - \lambda\end{aligned}\right\},\ \text{if}\ \left\{\begin{aligned}&z_1 \leq \lambda\,\gamma\, z_2 + \tau,\\ &z_2 \geq \lambda,\end{aligned}\right.\\
&\left.\begin{aligned}\hat{x}_1 &= \frac{z_1 - \lambda\,\gamma\, z_2 - \tau}{1 - \lambda^2\,\gamma^2}\\ \hat{x}_2 &= \frac{z_2 - \lambda\,\gamma\, z_1 - \tau\,\lambda}{1 - \lambda^2\,\gamma^2}\end{aligned}\right\},\ \text{if}\ \left\{\begin{aligned}&z_1 \geq \lambda\,\gamma\, z_2 + \tau,\\ &z_2 \geq \lambda\,\gamma\, z_1 + \tau,\end{aligned}\right.\\
&\left.\begin{aligned}\hat{x}_1 &= 0\\ \hat{x}_2 &= 0\end{aligned}\right\}\ \text{if}\ \left\{\begin{aligned}&z_1 \leq \lambda,\\ &z_2 \leq \lambda.\end{aligned}\right.
\end{aligned} \tag{11}$$

The regions defined on the right hand sides of (11) are denoted as R_i in Fig. 2a. On each R_i, the gradient of the threshold function is constant. We also remark that R_i's are determined

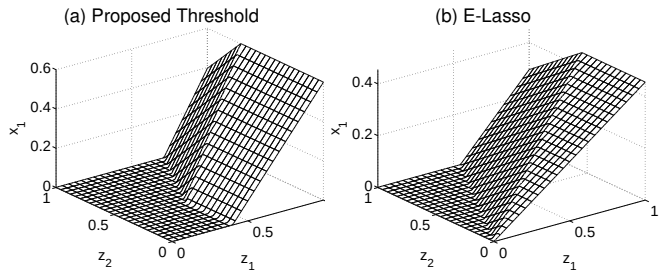

Fig. 3. The first component of the proposed threshold and E-Lasso.

by the two parameters λ and γ. R_4 is the deadzone (i.e., the collection of input vectors which are mapped to zero by the threshold function) for the bivariate threshold function and is determined by the weight λ. The first component of the threshold function (i.e., the mapping that takes $z = (z_1, z_2)$ to $\hat{x}_1$) is shown in Fig. 3a. Note that for this function, R_2 (on which $|z_2| \gg |z_1|$) is also a deadzone.

The proposed threshold function behaves quite differently than a threshold function derived from a separable penalty of the form '$p(x_1) + p(x_2)$'. If the penalty is separable, even if p is non-convex, the deadzone of the threshold function is rectangular (and hence $\hat{x}_1$ does not depend on z_2).

The relevant regions and the first component of the E-Lasso threshold are shown in Fig. 2b and Fig. 3b respectively. Note that, unlike the proposed threshold (which contains four different regions with constant gradient), the E-Lasso threshold has three regions where its gradient is constant. Also, unlike the proposed threshold, the E-Lasso threshold does not contain a deadzone that eliminates both components.

B. The Penalty and the Threshold Function on $\mathbb{R}^n$

We now consider the function $P_\gamma : \mathbb{R}^n \to \mathbb{R}_+$ defined in (2) and the associated threshold function. Unlike $\mathbb{R}^2$, it is not easy to express the threshold function in closed form in $\mathbb{R}^n$. Instead, we will derive a procedure to realize the threshold function. In order to justify the procedure, we will need to first study the properties of the penalty and the threshold functions.

We start with the penalty P_γ. This function is not convex but it becomes convex if we add a quadratic term. As mentioned before, such functions are called weakly-convex [32].

Definition 1. For $\gamma \geq 0$, a function f is said to be γ-weakly convex if

$$\frac{\gamma}{2}\|x\|_2^2 + f(x) \tag{12}$$

is convex.

Proposition 1. The function P_γ in (2) is γ-weakly convex. Consequently, the cost function $C_{\lambda,\gamma}(x|z)$ in (7) is strictly convex with respect to x if $\lambda\,\gamma < 1$.

Proof. To see the first claim, observe that,

$$\|x\|_1^2 = \|x\|_2^2 + 2\sum_{i=1}^{n-1}\sum_{m=i+1}^{n} |x_i\, x_m|. \tag{13}$$

Since $\|x\|_1^2$ is convex, this observation implies that the term in the parentheses in (2) is 1-weakly convex. Since $\|x\|_1$ is convex, it follows that P_γ is γ-weakly convex.

To see the strict convexity of $C_{\lambda,\gamma}$ with respect to x, note that it can be written as the sum of a convex function and

$$\frac{1}{2}\|x\|_2^2 + \lambda\,\gamma\left(\sum_{i=1}^{n-1}\sum_{m=i+1}^{n} |x_i\, x_m|\right). \tag{14}$$

But the function in (14) is strictly convex if $\lambda\,\gamma < 1$ and thus follows the claim. □

It also follows from Prop. 1 that, if $\lambda\,\gamma < 1$, then $T_{\lambda,\gamma}$ is well-defined thanks to the strict convexity of $C_{\lambda,\gamma}$.

In the following, we will derive two finite-terminating algorithms that realize $T_{\lambda,\gamma}$. For that, we first discuss the properties of $T_{\lambda,\gamma}$. We start by showing that $T_{\lambda,\gamma}(z)$ shrinks z towards zero and it is monotone in the sense that it preserves the ordering of $|z_i|$ with respect to i. More precisely, we have the following result.

Proposition 2. Let $\hat{x} = T_{\lambda,\gamma}(z)$, for $\lambda\,\gamma < 1$.
(a) If $z_i \geq 0$, then $z_i \geq \hat{x}_i \geq 0$. If $z_i \leq 0$, then $z_i \leq \hat{x}_i \leq 0$.
(b) If $|z_i| > |z_m|$, then $|\hat{x}_i| \geq |\hat{x}_m|$.
(c) If $|z_i| = |z_m|$, then $|\hat{x}_i| = |\hat{x}_m|$.

Proof. See Appendix A. □

We now derive an expression for $T_{\lambda,\gamma}$ using the optimality conditions. Notice that $C_{\lambda,\gamma}$ can be written as,

$$C_{\lambda,\gamma}(x|z) = \frac{1}{2}\|z\|_2^2 - \langle z, x\rangle + \frac{1-\lambda\,\gamma}{2}\|x\|_2^2 + \frac{\lambda\,\gamma}{2}\|x\|_1^2 + \lambda\|x\|_1. \tag{15}$$

Using this expression, the optimality conditions are found as,

$$z \in (1-\lambda\,\gamma)\hat{x} + \lambda\big(\gamma\|\hat{x}\|_1 + 1\big)\,\mathrm{sign}(\hat{x}), \tag{16}$$

where $\mathrm{sign}(\hat{x})$ is a set valued separable function of the vector $\hat{x}$, whose k^{th} component is given as,

$$\mathrm{sign}(\hat{x})_k = \begin{cases} \{1\}, & \text{if } \hat{x}_k > 0, \\ [-1,1], & \text{if } \hat{x}_k = 0, \\ \{-1\}, & \text{if } \hat{x}_k < 0. \end{cases} \tag{17}$$

In the following, we assume for simplicity that z_i's are non-negative and ordered, i.e., $z_1 \geq z_2 \geq \cdots \geq z_n \geq 0$. The general case can be recovered by a permutation of the vector components and changing signs, thanks to Prop. 2.

Prop. 2 implies that there is an index $k \in \{0, 1, \ldots, n\}$, such that $\hat{x}_i > 0$ if $i \leq k$ and $\hat{x}_i = 0$ if $i > k$. That is, k denotes the number of non-zeros in $\hat{x}$. For this special integer k, the optimality conditions can be written as,

$$z_i = (1-\lambda\,\gamma)\hat{x}_i + \lambda\left(\gamma\sum_{m=1}^{k}\hat{x}_m + 1\right), \text{if } i \leq k, \tag{18a}$$

$$z_i \leq \lambda\left(\gamma\sum_{m=1}^{k}\hat{x}_m + 1\right), \text{if } i > k. \tag{18b}$$

In order to find an expression for $\hat{x}_i$, let us define

$$\bar{z} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_k \end{bmatrix}, \quad \bar{x} = \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \vdots \\ \hat{x}_k \end{bmatrix}, \quad \mathbf{1}_k = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^k. \tag{19}$$

We can now express (18a) as

$$\bar{z} = (1 - \lambda\gamma)\,\bar{x} + \lambda\,\left(\gamma \mathbf{1}_k^T \bar{x} + 1\right)\,\mathbf{1}_k. \tag{20}$$

Multiplying both sides by $\mathbf{1}_k^T$ (noting $\mathbf{1}_k^T\,\mathbf{1}_k = k$) and rearranging, we have

$$\mathbf{1}_k^T \bar{x} = \frac{\mathbf{1}_k^T \bar{z} - k\,\lambda}{1 + (k-1)\lambda\gamma} \tag{21}$$

Therefore,

$$\lambda\,\left(\gamma \mathbf{1}_k^T \bar{x} + 1\right) = \frac{\lambda\,(1 - \lambda\,\gamma) + \lambda\,\gamma\,\sum_{j=1}^{k} z_j}{1 + (k-1)\,\lambda\,\gamma}. \tag{22}$$

The rhs of (22) will be of interest in the following. Let us therefore define, for each i,

$$h(i) = \frac{\lambda\,(1 - \lambda\,\gamma) + \lambda\,\gamma\,\sum_{m=1}^{i} z_m}{1 + (i-1)\,\lambda\,\gamma}. \tag{23}$$

Plugging the expression in (22) back into (18), we find the equivalent conditions

$$\hat{x}_i = (1 - \lambda\,\gamma)^{-1}\big(z_i - h(k)\big), \qquad \text{if } i \le k, \tag{24a}$$

$$z_i \le h(k), \qquad \text{if } i > k. \tag{24b}$$

Notice that the requirement $\hat{x}_i > 0$ for $i \le k$ implies that $z_i > h(k)$ for $i \le k$. The foregoing discussion is summarized in the following proposition.

Proposition 3. Let $\hat{x} = T_{\lambda,\gamma}(z)$, for $\lambda\,\gamma < 1$. Also, let k denote the number of non-zeros of $\hat{x}$. Then,

$$\hat{x} = (1 - \lambda\,\gamma)^{-1}\,\mathrm{soft}\big(z, h(k)\big), \tag{25}$$

where

$$h(k) = \frac{\lambda\,(1 - \lambda\,\gamma) + \lambda\,\gamma\,\sum_{m=1}^{k} |z_m|}{1 + (k-1)\,\lambda\,\gamma}, \tag{26}$$

(with the convention $\sum_{m=1}^{0} |z_m| = 0$).

We remark that the description of the threshold function in Prop. 3 is implicit because the integer k in (25), namely the number of non-zeros in $\hat{x}$, depends on $\hat{x}$. We next discuss how to determine the integer k. We will present two different search schemes for finding the correct value of k. The following lemma will be useful for that end.

Lemma 1. Suppose $z_1 \ge z_2 \ge \cdots z_n \ge 0$ and $\lambda\,\gamma < 1$. Let $h(i)$ be defined as in (23). Then,

(a) if $z_{i+1} > h(i)$, then $z_i > h(i-1)$,
(b) if $z_i \le h(i)$, then $z_{i+1} \le h(i+1)$.

Proof. See Appendix B. □

We can use this lemma to develop a procedure for determining k. It follows from the lemma that we can start from $k = 0$ and keep increasing k until $h(k) > z_{k+1}$. This procedure is summarized in Algorithm 1.

If k is suspected to be small, then this algorithm terminates quickly. In the worst case, the algorithm will execute the 'while' loop n times. If, however, n is large and k is not expected to be small, then a binary search for k might be computationally more suitable. The following discussion, that relies on Lemma 1 implies that such a binary search terminates.

Algorithm 1 Realization of $T_{\lambda,\gamma}$ – Linear Search for k

Require: $y \in \mathbb{R}^n$
  $z \leftarrow$ descending-sort($|y|$)
  $k \leftarrow 0$
  while $h(k) < z_{k+1}$ **do**
    $k \leftarrow k + 1$ {increment k}
  end while
  $\hat{x} \leftarrow (1 - \lambda\,\gamma)^{-1}\,\mathrm{soft}\big(y, h(k)\big)$.

Suppose we pick an arbitrary i and check the following conditions.

$$z_i > h(i), \tag{27a}$$

$$z_{i+1} \le h(i). \tag{27b}$$

Notice that since $z_i \ge z_{i+1}$, the conditions cannot be violated simultaneously. Now observe that

- If both conditions hold, the current guess of i is equal to the sought k.
- If (27a) holds and (27b) is violated, then by Lemma 1, k must be greater than i.
- If (27b) holds and (27a) is violated, then again by Lemma 1, k must be less than i.

These observations lead to an implementation of $T_{\lambda,\gamma}$ as in Algorithm 2. In contrast to the $O(n)$ complexity of Algorithm 1, this algorithm has $O\big(\log(n)\big)$ complexity.

Algorithm 2 Realization of $T_{\lambda,\gamma}$ – Binary Search for k

Require: $y \in \mathbb{R}^n$
  $z \leftarrow$ descending-sort($|y|$)
  flag $\leftarrow$ true
  if $z_1 < h(0)$ **then**
    $k \leftarrow 0$
    flag $\leftarrow$ false
  else if $z_n > h(n)$ **then**
    $k \leftarrow n$
    flag $\leftarrow$ false
  else
    $k_0 \leftarrow 0$ {left end of the interval}
    $k_1 \leftarrow n$ {right end of the interval}
  end if
  while flag **do**
    $k \leftarrow \lfloor (k_0 + k_1)/2 \rfloor$ {middle of the working interval}
    if $z_k > h(k)$ and $z_{k+1} \le h(k)$ **then**
      flag $\leftarrow$ false {correct value of k is found}
    else if $z_k \le h(k)$ **then**
      $k_1 \leftarrow k$ {update the right end}
    else
      $k_0 \leftarrow k$ {update the left end}
    end if
  end while
  $x \leftarrow (1 - \lambda\,\gamma)^{-1}\,\mathrm{soft}\big(y, h(k)\big)$.

C. Tuning the Parameters of the Threshold Function

Let us now discuss some special cases to better understand the role of the parameters λ and γ. As in the previous subsection, we will assume that $z_1 \geq \cdots \geq z_n \geq 0$ and $\hat{x} = T_{\lambda,\gamma}(z)$.

Observe first that $h(0) = \lambda$. If $z_i < \lambda$ for all i, then $\hat{x} = 0$. Thus the deadzone of the threshold function is a cube of width λ in $\mathbb{R}^n$.

Suppose now that $z_1 > \lambda$. In that case, we will definitely have $\hat{x}_1 > 0$. We find that

$$h(1) = \lambda + \lambda\gamma(z_1 - \lambda). \tag{28}$$

Notice that in order for $\hat{x}_2$ to be non-zero, the threshold that z_2 needs to exceed has increased from λ by an amount proportional to $(z_1 - \lambda)$. The higher z_1 is, the higher will be the new threshold. In fact, observe that as $\lambda\gamma \to 1$, the threshold converges to z_1. Since $z_2 \leq z_1$, we can therefore force only a single component to survive by choosing γ close to $1/\lambda$. When $z_2 < h(1)$, we find that,

$$\hat{x}_1 = z_1 - \lambda. \tag{29}$$

Thus the single surviving component is obtained by soft thresholding the largest component with λ.

The following proposition provides further information on how the potential thresholds $h(i)$ behave for arbitrary i.

Proposition 4. Suppose $z_1 \geq z_2 \geq \cdots z_n \geq 0$ and $\lambda\gamma < 1$. Let $h(i)$ be defined as in (23). If $z_{i+1} > h(i)$, then $z_{i+1} > h(i+1) > h(i)$.

Proof. See Appendix C. □

We know that if $z_{i+1} > h(i)$, then $h(i)$ is not the actual threshold and $k > i$. Prop. 4 implies that $h(k)$ is actually greater than $h(i)$, but it is bounded above by z_{i+1}. In fact, we can deduce from Prop. 4 that

$$z_1 \geq \cdots \geq z_k > h(k) \geq h(k-1) \geq \cdots \geq h(0) = \lambda. \tag{30}$$

In the case where the observations are purely noise, we would like to set $\hat{x}_i = 0$ for all i. This motivates choosing $\lambda = c\sigma$, where σ denotes the noise standard deviation and c is a constant around unity. Once we fix the value of λ, the number of non-zero components, k, and the threshold $h(k)$ will depend on γ (and z). The following proposition provides precise bounds on γ.

Proposition 5. Let $\hat{x} = T_{\lambda,\gamma}(z)$ where $\lambda\gamma < 1$ and $z_i \geq 0$ for all i. $\hat{x}_1 \geq \cdots \geq \hat{x}_k > 0$ and $\hat{x}_{k+1} = \cdots = \hat{x}_n = 0$ if and only if

$$\lambda\gamma > \frac{(z_{k+1} - \lambda)_+}{(z_{k+1} - \lambda)_+ + \sum_{i=1}^{k} (z_i - z_{k+1})}, \tag{31a}$$

$$\lambda\gamma < \frac{(z_k - \lambda)_+}{(z_k - \lambda)_+ + \sum_{i=1}^{k-1} (z_i - z_k)}. \tag{31b}$$

Proof. See Appendix D. □

Prop. 5 suggests that, if we would like to retain more components in the estimate $\hat{x}$, then we need to choose a small γ so that $\lambda\gamma$ is small. Also, if the signal is scaled by multiplying with a factor β, then the numbers on the rhs in (31) stay approximately the same. Therefore, the constant $\lambda\gamma$ controls the number of non-zeros, almost independently of the scale. To summarize, the following may serve as a guide for the selection of the parameters λ and γ.

- λ can be chosen proportional to the noise standard deviation.
- Once λ is chosen, γ can be selected independently of the scale of the signal but should satisfy $\lambda\gamma < 1$. The product $\lambda\gamma$ determines the number of non-zeros in the estimate. The higher it is, the lower the number of non-zeros will be.

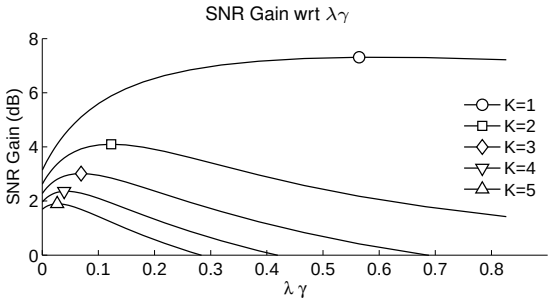

Fig. 4. Average SNR gains with respect to λ, γ, where λ is fixed and γ is varied. K denotes the number of non-zero components present in the clean signal.

In order to demonstrate the first point, we consider a simple experiment on synthetic signals. The desired signal is of length 10 and it has K non-zero components, where the non-zero values are obtained by sampling from a Gaussian distribution. We add Gaussian noise to this signal so that the observation SNR is 5 dB. We apply $T_{\lambda,\gamma}$ to the observation for a fixed $\lambda = \sigma/2$ and varying γ. We repeat the experiment for ten thousand trials to obtain an average figure. The average gain in SNR (dB) with respect to $\lambda\gamma$ is shown in Fig. 4. We see from the figure that the best $\lambda\gamma$ value decreases with increasing K. This is in line with Prop. 5, which suggests that in order for the reconstruction to have more non-zeros, the product $\lambda\gamma$ must be smaller.

D. Extension to $\mathbb{C}^n$

For $x \in \mathbb{C}^n$, we extend P_γ straightforwardly as,

$$P^c_\gamma(x) = \gamma \left(\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} |x_i\, x_j| \right) + \|x\|_1. \tag{32}$$

The threshold function is similarly defined as,

$$T^c_{\lambda,\gamma}(z) = \arg\min_{x \in \mathbb{C}^n} \frac{1}{2}\|x - z\|_2^2 + \lambda\, P^c_\gamma(x). \tag{33}$$

Fortunately, the threshold function derived for $\mathbb{R}^n$ applies for $\mathbb{C}^n$ with a little modification. The following observation is useful for showing that.

Lemma 2. Suppose $z \in \mathbb{C}^n$ and $\hat{x} = T^c_{\lambda,\gamma}(z)$. If $|\hat{x}_i| > 0$, then $\arg(\hat{x}_i) = \arg(z_i)$.

Proof. Suppose $\arg(\hat{x}_i) \neq \arg(z_i)$. Define $\tilde{x}$ such that $|\tilde{x}_i| = |\hat{x}_i|$ for all i and for $|\tilde{x}_i| > 0$, set $\arg(\tilde{x}_i) = \arg(z_i)$. Then,

$P^c_\gamma(\tilde{x}) = P^c_\gamma(\hat{x})$ but $\|z-\tilde{x}\|_2^2 < \|z-\hat{x}\|_2^2$, contradicting the fact that $\hat{x}$ minimizes the cost in (33). □

With the help of this lemma, we obtain an expression for $T^c_{\lambda,\gamma}$ in terms of $T_{\lambda,\gamma}$.

Proposition 6. Suppose $z \in \mathbb{C}^n$. Let $|z|$ denote the vector containing the magnitudes of the components of z. Let $\hat{x} = T^c_{\lambda,\gamma}(z)$ and $u = T_{\lambda,\gamma}(|z|)$. Then, $\hat{x}_k = u_k\, e^{j\arg(z_k)}$.

Proof. Notice that $|z_k| = z_k\, e^{-j\arg(z_k)}$. Using this observation, it can be shown by a change of variables that if $\tilde{x} = T^c_{\lambda,\gamma}(|z|)$, then $\tilde{x}_k = x_k\, e^{-j\arg(z_k)}$. Now since $\arg(|z_k|) = 0$, for all k, it follows by Lemma 2 that $\tilde{x}_k$ are real and non-negative. Thus, for the input $|z|$, we can restrict the minimization in (33) to $\mathbb{R}^n$. Thus $\tilde{x} = T_{\lambda,\gamma}(|z|) = u$ and the claim follows. □

It follows from this proposition that the threshold function on $\mathbb{C}^n$ can be realized by applying $T_{\lambda,\gamma}$ to the magnitudes of the input, followed by a correction of the argument of the complex number. For this reason, in the following, we will not differentiate between $T_{\lambda,\gamma}$ and $T^c_{\lambda,\gamma}$.

E. An Extension to Sub-Groups

We have so far considered a vector $x \in \mathbb{C}^n$ to comprise a group that is part of a larger signal. We now add an additional layer and consider subgroups of x to define a hybrid penalty, that can be used to complement $\ell_{2,1}$ norms. In this setting, we refer to x as a 'super-group'. Specifically, suppose x is partitioned into m non-overlapping sub-groups, i.e. $x = \begin{bmatrix} x^{(1)} & x^{(2)} & \dots & x^{(m)} \end{bmatrix}$. Also, let w denote the length-m vector whose i^{th} component is $w_i = \|x^{(i)}\|_2$. We define a hybrid penalty $\tilde{P}_\gamma(x)$ as,

$$\tilde{P}_\gamma(x) = P_\gamma(w). \tag{34}$$

Observe that,

$$\tilde{P}_\gamma(x) = \frac{\gamma}{2}\big(\|w\|_1^2 - \|w\|_2^2\big) + \|w\|_1 \tag{35}$$

$$= \frac{\gamma}{2}\left(\|x\|_{2,1}^2 - \|x\|_2^2\right) + \|x\|_{2,1}, \tag{36}$$

where $\|x\|_{2,1} = \sum_{i=1}^m \|x^{(i)}\|_2$. Therefore, $\tilde{P}_\gamma$ is γ-weakly convex. The threshold function of $\tilde{P}_\gamma$ is similarly defined as

$$\tilde{T}_{\lambda,\gamma}(z) = \arg\min_x \frac{1}{2}\|x-z\|_2^2 + \lambda\,\tilde{P}_\gamma(x). \tag{37}$$

Thanks to the weak-convexity of $\tilde{P}_{\lambda,\gamma}$, $\tilde{T}_{\lambda,\gamma}$ is well-defined for $\lambda\,\gamma < 1$. $\tilde{T}_{\lambda,\gamma}$ can be easily described using $T_{\lambda,\gamma}$, as follows (see also [19] for a discussion of convex multi-layer penalties).

Proposition 7. Suppose a vector z partitioned into m groups where the i^{th} group is denoted as $z^{(i)}$. Also, let w denote the length-m vector whose i^{th} component is $w_i = \|z^{(i)}\|_2$. Let $\hat{x} = \tilde{T}_{\lambda,\gamma}(z)$ and $\hat{w} = T_{\lambda,\gamma}(w)$. If we partition $\hat{x}$ into m groups similarly as z (with $\hat{x}^{(i)}$ denoting the i^{th} group), we have,

$$\hat{x}^{(i)} = \begin{cases} \dfrac{\hat{w}_i}{w_i}\, z^{(i)}, & \text{if } w_i > 0, \\ 0, & \text{if } w_i = 0. \end{cases} \tag{38}$$

Proof. See Appendix E. □

In words, this proposition implies that the orientation of $x^{(i)}$ is the same as that of $z^{(i)}$. To find the length of $\hat{x}^{(i)}$, i.e., $\|\hat{x}^{(i)}\|_2$, we apply the thresholding operator $T_{\lambda,\gamma}$ to w. We will consider an application of this penalty and threshold function in Sec. III for a convex denoising formulation.

III. Application-I : Convex Denoising with a Sparsifying Frame

We now consider the application of the proposed penalty in a denoising problem, when a sparsifying frame is given. By 'sparsifying frame', we mean a linear transform with a stable inverse (see [9] for a detailed discussion) which allows to represent the signal with high fidelity using a small number of transform domain coefficients. We will specifically seek a convex formulation for this problem.

A. A Convex Denoising Formulation

Let y be a noisy observation of a clean signal x_c for which a sparsifying frame is given. Let S and S^* denote the analysis and synthesis operators for the frame [9]. We assume that $S^*\,S = I$, i.e., the frame is Parseval [9]. We have two choices for formulating the denoising problem, namely synthesis and analysis prior formulations [14, 26]. The two behave quite differently under a non-convex penalty such as the one considered in this paper.

In the setting described above, the synthesis prior denoising formulation is,

$$\min_t \frac{1}{2}\|y - S^*\,t\|_2^2 + \lambda\,\mathbf{P}_\gamma(t). \tag{39}$$

If we denote the minimizer as $\hat{t}$, the denoised estimate is given as $\hat{x} = S^*\,\hat{t}$. In order to investigate convexity, let us rewrite the cost function in (39) as

$$\left[\frac{1}{2}\|y - S^*\,t\|_2^2 - \frac{\alpha}{2}\|t\|_2^2\right] + \left[\frac{\alpha}{2}\|t\|_2^2 + \lambda\,\mathbf{P}_\gamma(t)\right]. \tag{40}$$

Notice that the second term in square brackets in (40) is convex if $\lambda\,\gamma \le \alpha$. The Hessian of the first term in square brackets is $S\,S^* - \alpha I$. Thus, the first term is also convex if $S\,S^* \succeq \alpha I$. In that case, the problem in (39) will be convex. However, if the frame is overcomplete, S^*will have a non-trivial null-space and the condition $S\,S^* \succeq \alpha I$ is not satisfied for $\alpha > 0$. Therefore, we can guarantee the convexity of the synthesis prior problem only for $\gamma = 0$, for which $\mathbf{P}_\gamma$ is equivalent to an ℓ_1 norm. This leads us to consider the analysis prior formulation given as,

$$\min_x \frac{1}{2}\|y - x\|_2^2 + \lambda\,\mathbf{P}_\gamma(S\,x). \tag{41}$$

Proposition 8. Suppose S is the analysis operator of a Parseval frame. The problem in (41) is convex if $\lambda\,\gamma \le 1$.

Proof. Since the frame is Parseval, we have $\|Sx\|_2^2 = \|x\|_2^2$. Therefore the cost function in (41) can be written as,

$$\begin{aligned}&\frac{1}{2}\|y-x\|_2^2 - \frac{1}{2}\|x\|_2^2 + \frac{1}{2}\|Sx\|_2^2 + \lambda\,\mathbf{P}_\gamma(S\,x)\\ &\quad= \underbrace{\frac{1}{2}\|y\|_2^2 - \langle x, y\rangle}_{f_1(x)} + \underbrace{\frac{1}{2}\|Sx\|_2^2 + \lambda\,\mathbf{P}_\gamma(S\,x)}_{f_2(Sx)}. \end{aligned} \quad (42)$$

In (42), f_1 is an affine function and is therefore convex. The function $f_2(x)$ in (42) is convex when $\lambda\gamma \leq 1$, by Prop. 1. Since pre-composition with a linear operator preserves convexity [16], $\tilde{f}(x) = f_2(Sx)$ is also convex. Thus the cost function in (41) can be expressed as the sum of two convex functions and is therefore convex. □

1) The Douglas-Rachford Algorithm: In order to obtain a minimizer of (41), we adapt the Douglas-Rachford algorithm [22, 12, 10]. The Douglas-Rachford algorithm is suitable for minimization problems of the form

$$\min_t f(t) + g(t), \quad (43)$$

where both f and g are convex. The Douglas-Rachford iterations for such a problem are,

$$t^{k+1} = \frac{1}{2}t^k + \frac{1}{2}\big(2J_{\alpha f} - I\big)\big(2J_{\alpha g} - I\big)\,t^k, \quad (44)$$

where $\alpha > 0$ is a parameter and $J_{\alpha f}$, $J_{\alpha g}$ are proximity operators associated with f and g (as defined in (6)). The sequence t^k constructed in (44) converges to some t^* such that $J_{\alpha g}(t^*)$ minimizes (43).

2) Adapting the Douglas-Rachford Algorithm: The problem in (41) is not readily suitable for the application of the Douglas-Rachford algorithm because the Douglas-Rachford algorithm assumes that the cost function consists of the sum of two convex functions, and one of the functions in (41), namely $\lambda\,\mathbf{P}_\gamma(S\,x)$, is non-convex. We now transform the problem to write it in a suitable form.

Since $S^*\,S = I$, we have

$$\|x-y\|_2^2 = \|Sy - Sx\|_2^2. \quad (45)$$

Now if $\mathcal{R}(S)$ denotes the range of S, we can change variables and obtain a problem equivalent to (41) as,

$$\min_u \underbrace{\frac{1}{2}\|S\,y-u\|_2^2 + \lambda\,P_\gamma(u)}_{f(u)} + \underbrace{i_{\mathcal{R}(S)}(u)}_{g(u)}, \quad (46)$$

where $i_{\mathcal{R}(S)}$ is the indicator function of $\mathcal{R}(S)$ [16]. If u^* denotes a minimizer of (46), then $S^*\,u^*$ minimizes (41). In this formulation, both f and g are convex, provided that $\lambda\gamma \leq 1$. Thus the Douglas-Rachford algorithm is applicable for this splitting. We remark that in this setting, the proximity operator for $g = i_{\mathcal{R}(S)}$ is simply a projection onto $\mathcal{R}(S)$ (see e.g. [10]), which can be achieved by applying $S\,S^*$, thanks to the Parseval property of the frame. The proximity operator for f can be expressed in terms of the threshold function as follows.

$$\begin{aligned}&J_{\alpha f}(z)\\ &= \arg\min_u \frac{1}{2\alpha}\|z-u\|_2^2 + \frac{1}{2}\|Sy-u\|_2^2 + \lambda\,\mathbf{P}_\gamma(u) \quad (47a)\\ &= \arg\min_u \frac{1}{2}\left\|u - \frac{\alpha}{\alpha+1}\left(Sy + \frac{z}{\alpha}\right)\right\|_2^2 + \frac{\alpha}{\alpha+1}\lambda\,\mathbf{P}_\gamma(u) \quad (47b)\\ &= \mathbf{T}_{(\beta\,\lambda),\gamma}\left(\beta\,\left(Sy + \frac{z}{\alpha}\right)\right), \quad (47c)\end{aligned}$$

where $\beta = \dfrac{\alpha}{\alpha+1}$. We remark that in passing from (47a) to (47b), we discarded an additive term and removed a positive factor from the cost function (equalities remain valid because we are seeking the minimizer).

Resulting pseudocode for the Douglas-Rachford iterations for this problem is given in Algorithm 3.

Algorithm 3 Analysis Prior Denoising Algorithm

Initialize $\alpha > 0$, t.
Set $\beta \leftarrow (1+\alpha)^{-1}\,\alpha$
repeat
  $u \leftarrow S\,S^*\,t$
  $z \leftarrow \mathbf{T}_{\beta\,\lambda,\gamma}\Big(\beta\big(Sy + \alpha^{-1}(2u-t)\big)\Big)$
  $t \leftarrow z + t - u$
until convergence
$x^* \leftarrow S^*\,t$

B. Numerical Experiment

We now demonstrate how the denoising formulation/algorithm performs on an audio signal and compare it against formulations that employ different regularizers. The clean signal is a speech signal, sampled at 16 kHz, whose spectrogram is shown in Fig. 6a. We use the short-time Fourier transform (STFT) as the tight frame in this experiment. The window size is 60 ms (960 samples) and the hop-size is 15 ms (240 samples).

For regularization, we compare the ℓ_1 norm, E-Lasso, the $\ell_{2,1}$ norm and two different versions of the proposed penalty. Our aim in this experiment is two-fold. First, we demonstrate the difference between the proposed penalty, E-Lasso and the ℓ_1 norm. Second, we show that the proposed penalty can be used to complement the $\ell_{2,1}$ norm to obtain enhanced reconstructions.

In order to describe the group penalties, consider Fig. 5, which introduces notation for P_{EL} (E-Lasso penalty), P_γ (proposed) and $\ell_{2,1}$ norm. For P_{EL} and P_γ, we select the length along the time axis as $l = 1$ and the width along the frequency axis as $w = 16$. This covers a frequency bandwidth of 320 Hz. Our aim is to exploit the isolated appearance of the harmonics viewed along the frequency axis. In contrast, for the $\ell_{2,1}$ norm, we take $w = 1$ and $l = 8$. With these choices, we aim to collect the coefficients belonging to a harmonic into a single group. However, unlike P_{EL} and P_γ, regularization

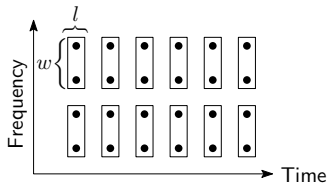

Fig. 5. Definition of the groups in the time-frequency lattice used in the denoising experiment in Section III-B. The parameters l and w denote the length along the time axis and the width along the frequency axis respectively.

with the $\ell_{2,1}$ norm does not specifically seek to isolate the harmonics like P_{EL} and P_γ. In order to obtain such an effect, we add an additional layer of grouping as depicted in Fig. 5 and use the penalty $\tilde{P}_\gamma$ introduced in Sec.II-E. We stack 16 neighboring groups along the frequency axis, used in the $\ell_{2,1}$ norm to define non-overlapping super-groups for $\tilde{P}_\gamma$.

We produce a noisy observation by adding noise (see Fig. 6b) consisting of ambient sounds recorded in a casino (machine sounds and crowd noise). Notice that energy of the ambient noise is not uniform over the frequencies and decreases with increasing frequency. Therefore, this noise can be considered pink. The input SNR is 5 dB. The spectrogram of the noisy signal is shown in Fig. 7a. For this observation, we perform denoising using the different regularizers in the analysis prior formulation, namely (41). The denoising algorithms for the different regularizers can be obtained by replacing $\mathbf{T}_{\lambda,\gamma}$ with the corresponding threshold functions in Algorithm 3. The value of the regularizer weight λ is selected by a sweep search for the ℓ_1 norm, E-Lasso and the $\ell_{2,1}$ norm[1] (output SNR maximizing value is chosen). For $P_{\lambda,\gamma}$, we set λ to be half the value of λ used for the ℓ_1 norm and perform a sweep search for γ subject to $\gamma < 1/\lambda$, to maximize the output SNR. For $\tilde{P}_\gamma$, we similarly set λ equal to half the value used for the $\ell_{2,1}$ norm and search for the best γ. The optimal choices of γ were found as 2.68 and 7.85 for P_γ and $\tilde{P}_\gamma$, respectively.

The resulting reconstructions are shown in Fig. 7. The output SNRs are 6.42 dB, 6.54 dB, 6.67 dB for ℓ_1 regularization, P_{EL} and P_γ respectively. Although the SNRs are close to each other, the reconstructions show different behaviors. Both ℓ_1 regularization and the proposed regularization have been successful in removing noise in the time-frequency regions with no activity. However since E-Lasso always keeps a component within a group, it has been less successful in suppressing noise in silent regions. We see that especially for higher frequencies, ℓ_1 regularization suppresses the harmonics of the speech signal. In contrast, the proposed penalty admits a smaller weight λ and is able to retain high frequency harmonics, while achieving a similar suppression of noise as ℓ_1 regularization.

For $\ell_{2,1}$ regularization and $\tilde{P}_\gamma$ regularization, the output SNRs are 6.09 and 7.55 dB, respectively. We observe that despite its lower SNR, $\ell_{2,1}$ preserves the harmonics better than the ℓ_1 norm. However, especially in the low frequency region, noise suppression is modest. While this can be overcome with a higher threshold λ for lower frequencies, such an approach

[1]A logarithmic search is performed for selecting λ. We refer to the available Matlab code for the sweep ranges.

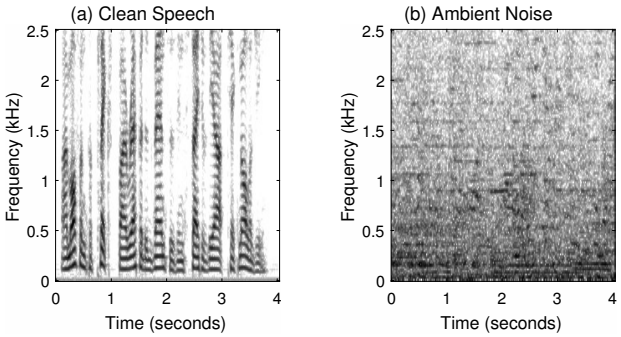

Fig. 6. Spectrograms of (a) the clean speech and (b) the noise signal used in the analysis prior denoising experiment.

requires choosing λ in a frequency dependent manner, which complicates the application. We observe however that $\tilde{P}_\gamma$ achieves suppression of noise in a wide frequency range, without having to tune parameters for each frequency separately. Specifically, noise within harmonics is eliminated similarly as in the reconstructions obtained by P_{EL} and P_γ. We also remark that specialized single-channel enhancement methods or more sophisticated penalties (such as weighted group penalties, like those in [19, 28]) can be used to achieve a superior performance than that of the proposed denoising method. Nevertheless, we think that the proposed penalty function can be used to complement or modify such alternatives as demonstrated here.

IV. Application II : Deconvolution

In a second application, we consider a sparse deconvolution problem. In order to be able to handle an arbitrary convolution operator, we forgo convexity and consider a non-convex formulation. We provide an algorithm for the provided formulation and discuss its convergence. We also compare the performance of the penalty/threshold function with a state-of-the-art iterative thresholding method.

A. A Non-Convex Formulation and a Convergent Algorithm

Consider a minimization formulation as

$$\min_x \left\{ D(x) = \frac{1}{2}\|y - H\,x\|_2^2 + \lambda\,\mathbf{P}_\gamma(x) \right\}, \tag{48}$$

where H denotes a convolution operator and $\mathbf{P}_\gamma$ is the proposed penalty. We remark that if H is not invertible, then $D(x)$ may not be convex unless $\gamma = 0$ (see the discussion in the begining of Sec. III-A). But if $\gamma = 0$, $\mathbf{P}_\gamma$ is the ℓ_1 norm. For this reason, unlike Section III, we will not restrict the cost function to be convex in this section. We employ the forward-backward splitting algorithm (FBS) [11, 10, 2] for obtaining a local minimizer of (48). FBS can be applied to cost functions that can be expressed as the sum of two functions, one of which is differentiable. The forward step updates the current estimate of the minimizer in the direction opposite to the gradient of the differentiable term. Following this update, the backward step consists of the application of the proximity operator of the remaining term. For the problem in (48), the data fidelity termis differentiable and the proximity operator

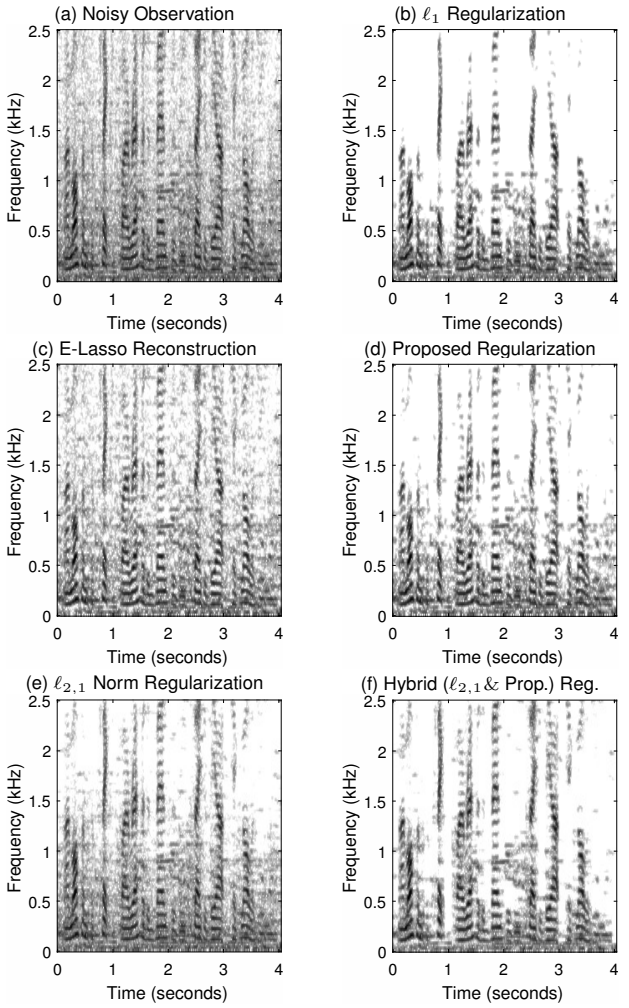

Fig. 7. Spectrograms of (a) the noisy signal (SNR = 5 dB) and the reconstructions using (b) the ℓ_1 norm (SNR = 6.42 dB), (c) the E-Lasso penalty (SNR = 6.54 dB), (d) the proposed penalty (SNR = 6.67 dB), (e) the $\ell_{2,1}$ norm (SNR = 6.09 dB), (f) the hybrid penalty in Sec. II-E, obtained by combining the $\ell_{2,1}$ norm with the proposed penalty (SNR = 7.55 dB).

of the penalty function $\lambda\,\mathbf{P}_\gamma(x)$ is available. The sequence constructed by FBS is,

$$x^{k+1} = \mathbf{T}_{(\alpha\lambda),\gamma}\big(x^k - \alpha H^T(Hx^k - y)\big). \tag{49}$$

We remark that $\mathbf{T}_{(\alpha\lambda),\gamma}$ is well-defined when $\alpha\,\lambda\,\gamma < 1$. This sets an upper bound on α. If, in addition, $\alpha < 1/\sigma(H)$, where $\sigma(H)$ denotes the spectral norm of H (which is equal to the square root of the largest eigenvalue of $H^T H$), it can also be shown using majorization-minimization techniques [21] that the sequence in (49) monotonically decreases the cost, i.e., $D(x^{k+1}) < D(x^k)$. Attouch et al. show in [2] that the algorithm (49) converges under the following additional conditions,

(i) P is a Kurdyka-Lojasiewicz function ([2], Defn. 2.4),
(ii) x^k's form a bounded sequence.

Both of these conditions are satisfied for our setup. We first remark that the proposed penalty function P_γ is continuous and for each orthant in $\mathbb{R}^n$, it can be expressed as a polynomial function. Therefore P_γ (therefore $\mathbf{P}_\gamma$) is semi-algebraic (see Defn. 2.1 in [2]) and hence is a Kurdyka-Lojasiewicz function (see the discussion at the end of Sec. 2.2 in [2]). Also, the proposed penalty is coercive, i.e., $\mathbf{P}_\gamma(x)$ increases without bound as $\|x\|_2$ increases. Therefore $D(x)$ is also coercive and any sequence that monotonically decreases $D(x)$ lies in a bounded set. To summarize, the following proposition is a corollary of Thm. 5.1 of [2].

Proposition 9. If $\alpha < \min(\sigma(H), 1/(\lambda\,\gamma))$, then x^k's in (49) decrease the cost $D(x)$ monotonically, and converge to a local minimizer.

B. Numerical Experiment

In exploration seismology [30, 25], the goal is to estimate an unknown reflectivity signal x from the observed seismic trace y, which is related to x as,

$$y = h * x + w, \tag{50}$$

where h represents the seismic wavelet and w denotes white noise. Denoting the convolution operator with h as H, we can use the formulation in (48) to estimate x from y. We will assume that the seismic wavelet, h, is known. Specifically, we experiment with the band-pass Ricker wavelet (dominant frequencies in the range $10 \sim 40$ Hz), sampled at $f_s = 300$ Hz, which is shown in Fig. 8a. We remark that the operator H used in the experiments is severely ill conditioned.

We use a synthetic sparse reflectivity signal for this experiment. The signal is selected by sampling a stochastic process where the probability of observing a non-zero at a specific sample is 0.1, provided that a non-zero has not occurred in the last 10 samples. Such a characteristic ensures that the spikes in the signal are fairly isolated. The value of the non-zero sample is obtained by sampling a normal distribution. Notice that, this process is not a sparse Bernoulli process but a Markov process due to the dependence on the past. The resulting synthetic x, of length $N = 512$ is shown in Fig. 8b. The observed seismic trace y, generated according to (50) is shown in Fig. 8c. We used zero-mean white Gaussian noise to produce y. The input SNR for this observation is 5 dB.

We compared the performance of the proposed algorithm with the sparse-group lasso formulation (SGL) [29] and the iterated p-shrinkage (IPS) algorithm [33], which was observed to perform very well for sparse deconvolution (see e.g. the comparisons in [27]).

SGL aims to achieve sparsity within groups and uses as few groups as possible for reconstruction. The SGL penalty is given as,

$$P_{\text{SGL}}(x) = \beta\,\|x\|_1 + (1-\beta)\sum_i \|x^{(i)}\|_2, \tag{51}$$

where $\beta \in (0,1)$ and $\|x^{(i)}\|_2$ denotes the ℓ_2 norm of the i^{th} group. Replacing $\mathbf{P}_\gamma$ with P_{SGL} in (48), we obtain the SGL formulation. We set $\beta = 0.95$ as in [29] and make a sweep search for selecting λ. The groups consist of neighboring intervals of length 8.

IPS employs a threshold function depending on two parameters, namely λ and p. The parameter p determines the shape of the threshold function and defines a family of functions that lie between soft ($p = 1$) and hard threshold ($p \to -\infty$).

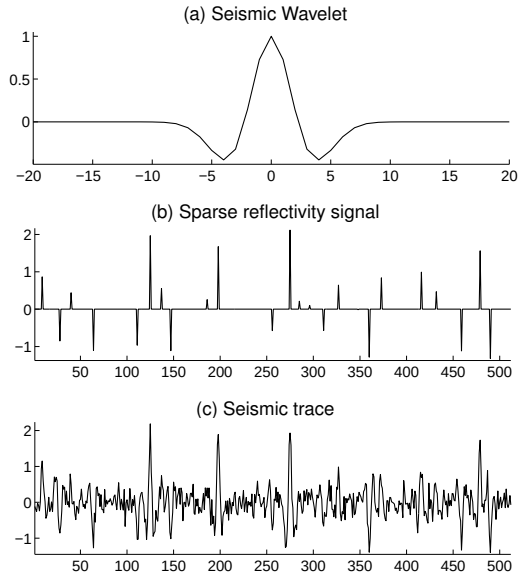

Fig. 8. Signals from the deconvolution experiment. (a) The seismic wavelet which is assumed to be known, (b) the sparse reflectivity signal, (c) observed noisy seismic trace.

TABLE I
SRER PERFORMANCE COMPARISON FOR DECONVOLUTION

SNR_{in}	Method	$\mathbb{E}(\text{SRER})$	$\sigma(\text{SRER})$
5 dB	SGL	10.08	0.92
	IPS	9.18	2.29
	Proposed	11.36	1.42
10 dB	SGL	14.58	0.91
	IPS	14.99	3.36
	Proposed	16.63	1.51
15 dB	SGL	19.41	0.87
	IPS	21.87	1.77
	Proposed	21.82	1.46
20 dB	SGL	24.19	0.87
	IPS	24.08	1.50
	Proposed	27.09	1.41

We selected $p = -1/2$, which gave fairly good results. The parameter λ is the threshold value and is selected with a sweep search.

Finally, for $\mathbf{P}_\gamma$, we use the same groups as SGL. We set $\gamma = 0.9/\lambda$ and select λ with a sweep search[2]. We remark that for the current setup, since the distance between two non-zeros of x is at least 10, each group of size 8 contains at most a single non-zero. This is the reason for choosing $\lambda\gamma$ close to unity. If multiple zeros were expected within a group, a lower value of γ would be more feasible.

We considered four different input SNRs (5, 10, 15, 20 dB) and evaluated the deconvolution performance using signal to reconstruction error ratio (SRER), $\|x\|_2/\|x - \hat{x}\|_2$, where $\hat{x}$ denotes the estimate. For each input SNR value, we repeat the experiment for 500 different noise realizations to obtain average and standard deviation statistics of the performance. We set α to be near the upper bound allowed in Prop. 9. We

[2]We refer the reader to the Matlab code for the sweep range.

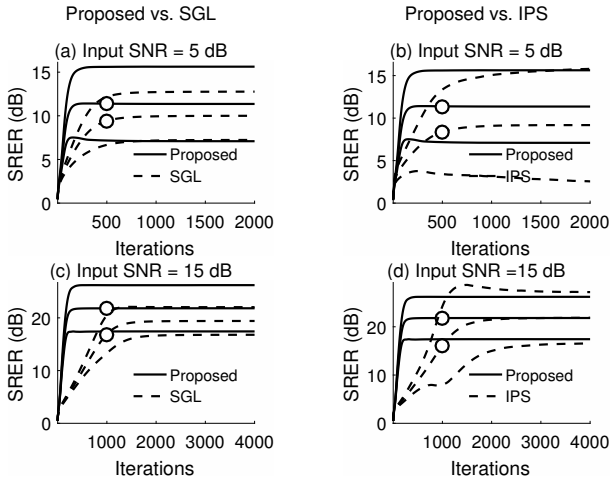

Fig. 9. Reliability plots comparing the proposed algorithm and iterative p-shrinkage (IPS) [33]. Each figure shows the signal to reconstruction error with respect to iterations. Solid and dashed lines belong to the proposed algorithm and IPS respectively. The means are marked with white circles. The other lines indicate three times the standard deviation from the means for each method.

remark that the proposed formulation and IPS are essentially non-convex formulations but we have seen that both algorithms converge in our experiments (as claimed by Prop. 9 and in [33]).

We observed an interesting trend with respect to different input SNRs. For low input SNRs, the proposed formulation performs better than the other two methods, in terms of average SRER. As the input SNR increases, the best SRER achieved with IPS sometimes surpasses those of the proposed formulation and SGL. However, the performance of IPS varies more with respect to different trials. On average, the proposed formulation performs better than both methods. Also, the proposed threshold function requires fewer iterations to fairly converge, although the iterations are computationally more costly. In order to visualize these, we show in Fig. 9 the SRER performance with respect to iterations for input SNRs 5 and 15 dB. Here, in addition to average SRER, three times the standard deviation of the SRER with respect to iterations is also shown. For the limits, the average SRER and the standard deviation of the SRER for the different algorithms are tabulated in Table I. In conclusion, the proposed penalty/formulation yields a better average SRER over SGL, which targets a similar property. However, its performance is less consistent with respect to SGL as seen by a comparison of $\sigma(\text{SRER})$. On the other hand, the proposed method is favorable compared to IPS. Especially for high input SNRs, the average SRER returned by IPS can be higher but stability is poorer. We think this is partly due to the clean sparse reflectivity signal which enforces a certain distance between the zeros. This property is taken into account by the proposed formulation but the IPS, which performs very well for arbitrary sparse signals, does not make use of this information.

V. CONCLUSION

We proposed a group separable penalty function suitable for signals showing a sparse behavior both across and within groups of coefficients. We derived an associated threshold

function $\mathbf{T}_{\lambda,\gamma}$ and studied how it behaves as its parameters vary. We noted that λ can be chosen to be proportional to the noise standard deviation and for fixed λ, the parameter γ determines how many non-zero coefficients are expected in each group. Specifically, as $\gamma \to 1/\lambda$, we showed that in each group, there remains at most one non-zero coefficient (when the largest coefficient exceeds the threshold λ).

We think that the proposed penalty/threshold would be of interest in several areas, such as EEG source localization [15], seismic deconvolution [30, 25], audio processing, specifically decomposing audio signals into transient and tonal components [20, 6], low-rank matrix recovery [7, 23] with a bound on the rank. We hope to consider such applications in future work.

Acknowledgement

We thank Pavel Rajmic, Brno University of Technology, Czech Republic, for discussions and comments.

APPENDIX A
PROOF OF PROP. 2

Recall that $\hat{x} = T_{\lambda,\gamma}(z)$ is the minimizer of $C_{\lambda,\gamma}(x|z)$ in (7) with respect to x.

(a) Let $z_i \geq 0$.
Assume $\hat{x}_i > z_i$. Define a new vector x^* as

$$x_i^* = \max\big(z_i - (\hat{x}_i - z_i), 0\big), \tag{52}$$
$$x_m^* = \hat{x}_m, \text{ if } m \neq i. \tag{53}$$

Then, $\|x^* - z\|_2^2 \leq \|\hat{x} - z\|_2^2$ and $P(x^*) < P(\hat{x})$. Therefore, $C_{\lambda,\gamma}(x^*|z) < C_{\lambda,\gamma}(\hat{x}|z)$. In words, x^* achieves a strictly lower cost than $\hat{x}$, which is a contradiction. Thus we must have, $\hat{x}_i \leq z_i$.
Assume $\hat{x}_i < 0$. Define a new vector x^* as

$$x_i^* = 0, \tag{54}$$
$$x_m^* = \hat{x}_m, \text{ if } m \neq i. \tag{55}$$

Then, $\|x^* - z\|_2^2 < \|\hat{x} - z\|_2^2$ and $P(x^*) < P(\hat{x})$. Therefore, x^* achieves a strictly lower cost than $\hat{x}$, which is a contradiction. Thus we must have, $\hat{x}_i \geq 0$.
The second part of the claim follows similarly.

(b) By part (a), we can assume without loss of generality that z has non-negative components. Assume $z_i > z_m \geq 0$. Note that by part (a), $\hat{x}_i \geq 0$, $\hat{x}_m \geq 0$. Suppose now that $\hat{x}_i < \hat{x}_m$. Define a new vector x^* as

$$x_i^* = \hat{x}_m, \tag{56}$$
$$x_m^* = \hat{x}_i, \tag{57}$$
$$x_l^* = \hat{x}_l, \text{ if } l \neq i \text{ or } l \neq m. \tag{58}$$

Then, $P_\gamma(x^*) = P_\gamma(\hat{x})$. We obtain after some algebraic manipulations that

$$\frac{1}{2}\left(\|\hat{x} - z\|_2^2 - \|x^* - z\|_2^2\right) = -\left(z_m - z_i\right)\left(\hat{x}_m - \hat{x}_i\right) > 0. \tag{59}$$

It thus follows $C_{\lambda,\gamma}(x^*|z) < C_{\lambda,\gamma}(\hat{x}|z)$, which is a contradiction. Thus $\hat{x}_i \geq \hat{x}_m$.

(c) Without loss of generality, assume z has non-negative components. Assume also that $z_i = z_m \geq 0$. By part (a), we will have $\hat{x}_i \geq 0$, $\hat{x}_m \geq 0$. Suppose now that $\hat{x}_i > \hat{x}_m \geq 0$. Set $a = (\hat{x}_i + \hat{x}_m)/2$ and define a new vector x^* as

$$x_i^* = x_m^* = a, \tag{60}$$
$$x_l^* = \hat{x}_l, \text{ if } l \neq i \text{ or } l \neq m. \tag{61}$$

Observe that $\|\hat{x}\|_1 = \|x^*\|_1$ and $\|\hat{x}\|_2 > \|x^*\|_2$. Using $z_i = z_m$, we first note,

$$\frac{1}{2}\left(\|\hat{x} - z\|_2^2 - \|x^* - z\|_2^2\right) = \frac{1}{2}\left(\|\hat{x}\|_2^2 - \|x^*\|_2^2\right). \tag{62}$$

Also, since $P_\gamma(\cdot) = \frac{\gamma}{2}(\|\cdot\|_1^2 - \|\cdot\|_2^2) + \|\cdot\|_1$, we find

$$P_\gamma(\hat{x}) - P_\gamma(x^*) = \frac{\gamma}{2}(\|x^*\|_2^2 - \|\hat{x}\|_2^2). \tag{63}$$

Using these, we obtain,

$$C_{\lambda,\gamma}(\hat{x}|z) - C_{\lambda,\gamma}(x^*|z) = \frac{1-\lambda\gamma}{2}(\|\hat{x}\|_2^2 - \|x^*\|_2^2) > 0. \tag{64}$$

Thus x^* achieves a lower cost than $\hat{x}$, which is a contradiction. Therefore $\hat{x}_i \leq \hat{x}_m$. Changing the roles of the indices m and i, we must also have $\hat{x}_m \leq \hat{x}_i$. Therefore, $\hat{x}_i = \hat{x}_m$.

APPENDIX B
PROOF OF LEMMA 1

(a) Since z_i's are ordered, the assumption $z_{i+1} > h(i)$ implies

$$z_i \geq z_{i+1} > \frac{\lambda\,(1-\lambda\,\gamma) + \lambda\,\gamma\,\sum_{j=1}^{i} z_j}{1 + (i-1)\,\lambda\,\gamma}. \tag{65}$$

This in turn implies

$$z_i(1 + (i-1)\,\lambda\,\gamma) > \lambda\,(1-\lambda\,\gamma) + \lambda\,\gamma\,\sum_{j=1}^{i} z_j. \tag{66}$$

Subtracting $\lambda\,\gamma\,z_i$ from both sides and rearranging, we obtain

$$z_i > \frac{\lambda\,(1-\lambda\,\gamma) + \lambda\,\gamma\,\sum_{j=1}^{i-1} z_j}{1 + (i-2)\,\lambda\,\gamma} = h(i-1). \tag{67}$$

(b) The proof of this part is similar. Since $z_{i+1} \leq z_i$, the assumption $z_i \leq h(i)$ implies

$$z_{i+1}(1 + (i-1)\,\lambda\,\gamma) \leq \lambda\,(1-\lambda\,\gamma) + \lambda\,\gamma\,\sum_{j=1}^{i} z_j. \tag{68}$$

Adding $\lambda\,\gamma\,z_{i+1}$ to both sides and rearranging, we obtain

$$z_{i+1} \leq \frac{\lambda\,(1-\lambda\,\gamma) + \lambda\,\gamma\,\sum_{j=1}^{i+1} z_j}{1 + i\,\lambda\,\gamma} = h(i+1). \tag{69}$$

APPENDIX C
PROOF OF PROP. 4

Assume $z_{i+1} > h(i)$. This inequality implies, by the definition of $h(i)$ in (23) that

$$\lambda\,(1-\lambda\,\gamma)+\lambda\,\gamma\,\sum_{j=1}^{i} z_j < (1+(i-1)\,\lambda\,\gamma)\;z_{i+1}. \quad (70)$$

We first show that $z_{i+1} > h(i+1)$. Using (70) in $h(i+1)$, we find

$$h(i+1) = \frac{\lambda\,(1-\lambda\,\gamma)+\lambda\,\gamma\,\sum_{j=1}^{i} z_j + \lambda\,\gamma z_{i+1}}{1+i\,\lambda\,\gamma} \quad (71)$$

$$< \frac{(1+(i-1)\,\lambda\,\gamma)\;z_{i+1}+\lambda\,\gamma z_{i+1}}{1+i\,\lambda\,\gamma} \quad (72)$$

$$= \frac{(1+i\,\lambda\,\gamma)\;z_{i+1}}{1+i\,\lambda\,\gamma} \quad (73)$$

$$= z_{i+1}. \quad (74)$$

Let us now show that $h(i+1) > h(i)$. Notice that for positive a, b, c, d,

$$\frac{a+c}{b+d} > \frac{a}{b}, \quad (75)$$

if and only if $ad < bc$. Now if we set

$$a = \lambda\,(1-\lambda\,\gamma)+\lambda\,\gamma\,\sum_{j=1}^{i} z_j, \quad (76)$$

$$b = (1+(i-1)\,\lambda\,\gamma)\,, \quad (77)$$

$$c = \lambda\,\gamma\,z_{i+1}, \quad (78)$$

$$d = \lambda\,\gamma, \quad (79)$$

then $h(i) = a/b$ and $h(i+1) = (a+c)/(b+d)$. But we have, by (70)

$$\frac{a\,d}{b\,c} = \frac{\lambda\,(1-\lambda\,\gamma)+\lambda\,\gamma\,\sum_{j=1}^{i} z_j}{(1+(i-1)\,\lambda\,\gamma)\;z_{i+1}} < 1. \quad (80)$$

Thus $h(i+1) > h(i)$.

APPENDIX D
PROOF OF PROP. 5

We have that $\hat{x}_1 \geq \cdots \geq \hat{x}_k > 0$ and $\hat{x}_{k+1} = \cdots = \hat{x}_n = 0$ if and only if $z_k > h(k) > z_{k+1}$. Plugging in the definition of $h(k)$ in (23), into the inequality $z_k > h(k)$, we obtain

$$z_k\left(1+(k-1)\lambda\,\gamma\right) > \lambda\,(1-\lambda\gamma)+\lambda\gamma\sum_{i=1}^{k} z_i. \quad (81)$$

Redistributing z_k's we can write,

$$z_k\,(1-\lambda\,\gamma) > \lambda\,(1-\lambda\gamma)+\lambda\gamma\sum_{i=1}^{k-1}(z_i-z_k). \quad (82)$$

This is equivalent to

$$(z_k-\lambda)\,(1-\lambda\,\gamma) > \lambda\gamma\sum_{i=1}^{k-1}(z_i-z_k). \quad (83)$$

Dividing both sides by the positive $(z_k-\lambda)\,\lambda\,\gamma$, we find,

$$\frac{1}{\lambda\,\gamma}-1 > \frac{\sum_{i=1}^{k-1}(z_i-z_k)}{z_k-\lambda}. \quad (84)$$

Rearranging this equation, we obtain (31b). (31a) can be shown similarly.

APPENDIX E
PROOF OF PROP. 7

In addition to the notation introduced in the proposition statement, let us also define $\tilde{w}$ to be length-$m$vector such that $\tilde{w}_i = \|\hat{x}^{(i)}\|_2$. We first observe that if $z^{(i)} = 0$, then $\hat{x}^{(i)} = 0$, for otherwise we could reduce the cost by setting $\hat{x}^{(i)} = 0$. Let us denote

$$\partial P_\gamma(\tilde{w}) = (\gamma\|\tilde{w}\|_1+1)\;\mathrm{sign}(\tilde{w}) - \gamma\tilde{w}, \quad (85)$$

where 'sign' is the set valued mapping defined in (17). Then, the optimality conditions for (37) imply that

$$0 \in \hat{x}^{(i)} - z^{(i)} + \lambda\left(\partial P_\gamma(\tilde{w})\right)_i u^{(i)}, \text{ for } i = 1,2,\ldots,m, \quad (86)$$

where $u^{(i)}$ is a unit norm vector such that $\langle \hat{x}^{(i)}, u^{(i)}\rangle = \|\hat{x}^{(i)}\|_2$. That is, if $\hat{x}^{(i)} \neq 0$, then $\hat{x}^{(i)} = \|\hat{x}^{(i)}\|_2\,u^{(i)}$. This in turn implies that if $\hat{x}^{(i)} \neq 0$, then since $z^{(i)} \neq 0$ (by the observation noted above), we must also have $z^{(i)} = \|z^{(i)}\|_2 u^{(i)}$, or $\langle z^{(i)}, u^{(i)}\rangle = \|z^{(i)}\|_2$. Taking inner products with $u^{(i)}$ in (86), we thus find,

$$0 \in \tilde{w}_i - w_i + \lambda\left(\partial P_\gamma(\tilde{w})\right)_i, \text{ for } i = 1,2,\ldots,m. \quad (87)$$

But these are the optimality conditions for the problem of minimizing $C_{\lambda,\gamma}(\cdot|w)$. Therefore, $\tilde{w} = T_{\lambda,\gamma}(w)$ and the claim follows.

REFERENCES

[1] D. Angelosante, G. B. Giannakis, and N. D. Sidiropoulos. Sparse parametric models for robust nonstationary signal analysis. *IEEE Signal Processing Magazine*, 30(6):64–73, November 2013.

[2] H. Attouch, J. Bolte, and B. F. Svaiter. Convergence of descent methods for semi-algebraic and tame problems: Proximal algorithms, forward-backward splitting, and regularized Gauss-Seidel methods. *Mathematical Programming*, 137(1):91–129, February 2013.

[3] P. Balazs, M. Dörfler, M. Kowalski, and B. Torrésani. Adapted and adaptive linear time-frequency representations: A synthesis point of view. *IEEE Signal Processing Magazine*, 30(6):20–31, November 2013.

[4] H. H. Bauschke and P. L. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*. Springer, 2011.

[5] İ. Bayram. Mixed-norms with overlapping groups as signal priors. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Proc. (ICASSP)*, 2011.

[6] İ. Bayram and Ö. D. Akyıldız. Primal-dual algorithms for audio decomposition using mixed norms. *Signal, Image and Video Processing*, 8(1):95–110, January 2014.

[7] E. J. Candés and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, December 2009.

[8] P.-Y. Chen and I. W. Selesnick. Translation-invariant shrinkage/thresholding of group sparse signals. *Signal Processing*, 94:476–489, January 2014.

[9] O. Christensen. *An Introduction to Frames and Riesz Bases*. Birkhäuser, 2003.

[10] P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In H. H. Bauschke, R. S. Burachik, P. L. Combettes, V. Elser, D. R. Luke, and H. Wolkowicz, editors, *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*. Springer, New York, 2011.

[11] P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *SIAM Journal on Multiscale Modelling and Simulation*, 4(4):1168–1200, November 2005.

[12] J. Eckstein and D. P. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(3):293–318, 1992.

[13] M. El Anbari and A. Mkhadri. Penalized regression combining the l_1 norm and a correlation based penalty. *Sankhya B*, 76(1):82–102, 2014.

[14] M. Elad, P. Milanfar, and R. Rubinstein. Analysis versus synthesis in signal priors. *Inverse Problems*, 23(3):947–968, June 2007.

[15] A. Gramfort and M. Kowalski. Improving M/EEG source localization with an inter-condition sparse prior. In *Proc. International Symposium on Biomedical Imaging (ISBI)*, 2009.

[16] J.-B. Hiriart-Urruty and C. Lemaréchal. *Fundamentals of Convex Analysis*. Springer, 2004.

[17] L. Jacob, G. Obozinsky, and J. P. Vert. Group lasso with overlap and graph lasso. In *Proc. Int. Conf. Machine Learning (ICML)*, 2009.

[18] M. Kowalski. Sparse regression using mixed norms. *Applied and Computational Harmonic Analysis*, 27(3):303–324, November 2009.

[19] M. Kowalski, K. Siedenburg, and M. Dörfler. Social sparsity! Neighborhood systems enrich structured shrinkage operators. *IEEE Transactions on Signal Processing*, 61(10):2498–2511, May 2013.

[20] M. Kowalski and B. Torrésani. Sparsity and persistence: Mixed norms provide simple signal models with dependent coefficients. *Signal, Image and Video Processing*, 3(3):251–264, 2009.

[21] K. Lange. *Optimization*. Springer, 2004.

[22] P. L. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, 16(6):964–979, 1979.

[23] A. Parekh and I. W. Selesnick. Enhanced low-rank matrix approximation. *IEEE Signal Processing Letters*, 23(4):493–497, April 2016.

[24] N. Rao, R. Nowak, C. Cox, and T. Rogers. Classification with the sparse group lasso. *IEEE Transactions on Signal Processing*, 64(2):448–463, January 2016.

[25] A. Repetti, M. Q. Pham, L. Duval, É. Chouzenoux, and J. C. Pesquet. Euclid in a taxicab: Sparse blind deconvolution with smoothed ℓ_1/ℓ_2 regularization. *IEEE Signal Processing Letters*, 22(5):539–543, May 2015.

[26] I. Selesnick and M. A. T. Figueiredo. Signal restoration with overcomplete wavelet transforms : Comparison of analysis and synthesis priors. In *Proceedings of SPIE (Wavelets XIII)*, 2009.

[27] I. W. Selesnick and İ. Bayram. Enhanced sparsity by non-separable regularization. *IEEE Transactions on Signal Processing*, 64(9):2298–2313, May 2016.

[28] K. Siedenburg and M. Dörfler. Structured sparsity for audio signals. In *Proc. Int. Conf. on Digital Audio Effects (DAFx)*, 2011.

[29] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani. A sparse-group lasso. *Journal of Computational and Graphical Statistics*, 22(2):231–245, 2013.

[30] A. K. Takahata, E. Z. Nadalin, R. Ferrari, L. T. Duarte, R. Suyama, R. R. Lopes, J. M. T. Romano, and M. Tygel. Unsupervised processing of geophysical signals: A review of some key aspects of blind deconvolution and blind source separation. *IEEE Signal Processing Magazine*, 29(4):27–35, July 2012.

[31] G. Tutz and J. Ulbricht. Penalized regression with correlation-based penalty. *Statistics and Computing*, 19(3):239–253, 2009.

[32] J.-P. Vial. Strong and weak convexity of sets and functions. *Mathematics of Operations Research*, 8:231–259, May 1983.

[33] J. Woodworth and R. Chartrand. Compressed sensing recovery via nonconvex shrinkage penalties. *Inverse Problems*, 32(7):075004, 2016.

[34] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society B*, 68(1):49–67, 2006.

[35] Y. Zhou, R. Jin, and S. Hoi. Exclusive lasso for multi-task feature selection. In *Proc. Int. Conf. Artificial Intelligence and Statistics*, 2010.

[36] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society B*, 67(2):301–320, April 2005.