S-SPADE Done Right:
Detailed Study of the Sparse Audio Declipper Algorithms

Pavel Záviška, Ondřej Mokrý, Pavel Rajmic[*]

October 30, 2018

Abstract: This technical report shows and discusses in detail how Sparse Audio Declipper (SPADE) algorithms are derived from the signal model using the ADMM approach. The analysis version (A-SPADE) of Kitić et. al. (LVA/ICA 2015) is derived and justified. The synthesis version (S-SPADE) of the same research team is shown to solve a different optimization task than intended. This issue is corrected in this report, leading to the new S-SPADE algorithm which is in line to A-SPADE.

1 Alternating Direction Method of Multipliers (ADMM)

We start with the brief review of ADMM [1]. ADMM is able to solve problems of the form

$$\min_{\mathbf{x}} f(\mathbf{x}) + g(A\mathbf{x}), \tag{1}$$

where $\mathbf{x} \in \mathbb{C}^N$ and $A : \mathbb{C}^N \to \mathbb{C}^P$ is a linear operator. We assume f, g real convex functions of (possibly complex) variables. Problem (1) can be reformulated as follows:

$$\min_{\mathbf{x},\mathbf{z}} f(\mathbf{x}) + g(\mathbf{z}) \quad \text{s.t.} \quad A\mathbf{x} - \mathbf{z} = 0. \tag{2}$$

To solve optimization task (2) we form the Augmented Lagrangian as

$$L_\rho(\mathbf{x}, \mathbf{y}, \mathbf{z}) = f(\mathbf{x}) + g(\mathbf{z}) + \mathbf{y}^\top(A\mathbf{x} - \mathbf{z}) + \frac{\rho}{2}\|A\mathbf{x} - \mathbf{z}\|_2^2, \tag{3}$$

where $\rho > 0$ is called the *penalty parameter*. ADMM consists of three steps:

$$\mathbf{x}^{(i+1)} = \underset{\mathbf{x}}{\operatorname{argmin}}\, L_\rho\left(\mathbf{x}, \mathbf{z}^{(i)}, \mathbf{y}^{(i)}\right) \tag{4a}$$

$$\mathbf{z}^{(i+1)} = \underset{\mathbf{z}}{\operatorname{argmin}}\, L_\rho\left(\mathbf{x}^{(i+1)}, \mathbf{z}, \mathbf{y}^{(i)}\right) \tag{4b}$$

$$\mathbf{y}^{(i+1)} = \mathbf{y}^{(i)} + \rho\left(A\mathbf{x}^{(i+1)} + \mathbf{z}^{(i+1)}\right). \tag{4c}$$

It is possible to convert ADMM to the so-called *scaled form*, which is often more convenient; this is done by defining the residual $\mathbf{r} = A\mathbf{x} - \mathbf{z}$. In such a case, the last two terms of the Augmented Lagrangian in (3) can be rewritten as

$$\mathbf{y}^\top \mathbf{r} + \frac{\rho}{2}\|\mathbf{r}\|_2^2 = \frac{\rho}{2}\left\|\mathbf{r} + \frac{1}{\rho}\mathbf{y}\right\|_2^2 - \frac{1}{2\rho}\|\mathbf{y}\|_2^2 = \frac{\rho}{2}\|\mathbf{r} + \mathbf{u}\|_2^2 - \frac{\rho}{2}\|\mathbf{u}\|_2^2, \tag{5}$$

where $\mathbf{u}$ is a *scaled dual variable* such that $\mathbf{u} = \mathbf{y}/\rho$ (see Remark 1 for the proof of Eq. (5)).

After the above manipulation, the Augmented Lagrangian in the scaled form is

$$L_\rho(\mathbf{x}, \mathbf{u}, \mathbf{z}) = f(\mathbf{x}) + g(\mathbf{z}) + \frac{\rho}{2}\|\mathbf{r} + \mathbf{u}\|_2^2 - \frac{\rho}{2}\|\mathbf{u}\|_2^2. \tag{6}$$

The scaled version of the ADMM is expressed as

$$\mathbf{x}^{(i+1)} = \underset{\mathbf{x}}{\operatorname{argmin}}\left(f(\mathbf{x}) + \frac{\rho}{2}\|A\mathbf{x} - \mathbf{z}^{(i)} + \mathbf{u}^{(i)}\|_2^2 \right) \tag{7a}$$

$$\mathbf{z}^{(i+1)} = \underset{\mathbf{z}}{\operatorname{argmin}}\left(g(\mathbf{z}) + \frac{\rho}{2}\|A\mathbf{x}^{(i+1)} - \mathbf{z} + \mathbf{u}^{(i)}\|_2^2 \right) \tag{7b}$$

$$\mathbf{u}^{(i+1)} = \mathbf{u}^{(i)} + A\mathbf{x}^{(i+1)} - \mathbf{z}^{(i+1)}. \tag{7c}$$

Note that in minimizing $L_\rho(\mathbf{x}, \mathbf{y}, \mathbf{z})$ over $\mathbf{x}$ in the first step (7a), it was possible to omit $g(\mathbf{z})$. This term is not dependent on $\mathbf{x}$, thus it does not play any role in finding the argument of the minima of $L_\rho(\mathbf{x}, \mathbf{y}, \mathbf{z})$. The same argumentation applies to $\frac{\rho}{2}\|\mathbf{u}\|_2^2$. Eq. (7b) is obtained the same way.

Remark 1. *The second equality in Eq. (5) is just a substitution. The first one can be proven (right to left) using the following, assuming $\mathbf{r}$ and $\mathbf{y}$ are real for the moment:*

$$\begin{aligned}
\frac{\rho}{2}\|\mathbf{r} + \frac{1}{\rho}\mathbf{y}\|_2^2 - \frac{1}{2\rho}\|\mathbf{y}\|_2^2 &= \frac{\rho}{2}\langle \mathbf{r} + \frac{1}{\rho}\mathbf{y}, \mathbf{r} + \frac{1}{\rho}\mathbf{y}\rangle - \frac{1}{2\rho}\langle \mathbf{y}, \mathbf{y}\rangle \\
&= \frac{\rho}{2}\left(\langle \mathbf{r}, \mathbf{r}\rangle + \langle \mathbf{r}, \frac{1}{\rho}\mathbf{y}\rangle + \langle \frac{1}{\rho}\mathbf{y}, \mathbf{r}\rangle + \langle \frac{1}{\rho}\mathbf{y}, \frac{1}{\rho}\mathbf{y}\rangle \right) - \frac{1}{2\rho}\langle \mathbf{y}, \mathbf{y}\rangle \\
&= \frac{\rho}{2}\|\mathbf{r}\|_2^2 + \frac{1}{2}\mathbf{r}^\top\mathbf{y} + \frac{1}{2}\mathbf{y}^\top\mathbf{r} \\
&= \frac{\rho}{2}\|\mathbf{r}\|_2^2 + \mathbf{y}^\top\mathbf{r}.
\end{aligned} \tag{8}$$

In the case of complex variables, we need to guarantee the Augmented Lagrangian to be real (and therefore the minimization is legitimate). For this reason, the Augmented Lagrangian in Eq. (3) would need to be alternatively defined as

$$L_\rho(\mathbf{x}, \mathbf{y}, \mathbf{z}) = f(\mathbf{x}) + g(\mathbf{z}) + \mathbf{y}^\top \begin{bmatrix} \Re(A\mathbf{x} - \mathbf{z}) \\ \Im(A\mathbf{x} - \mathbf{z}) \end{bmatrix} + \frac{\rho}{2}\left\| \begin{bmatrix} \Re(A\mathbf{x} - \mathbf{z}) \\ \Im(A\mathbf{x} - \mathbf{z}) \end{bmatrix} \right\|_2^2, \tag{9}$$

now with $\mathbf{y} \in \mathbb{R}^{2P}$. Then, (8) can be used again to prove Eq. (5), leading to Augmented Lagrangian in the form (6)—the only difference is that $\mathbf{r}, \mathbf{u} \in \mathbb{R}^{2P}$ instead of $\mathbb{C}^P$. In the implementation, this difference can be ignored and complex variables $\mathbf{r}$ and $\mathbf{u}$ can be directly used since it holds

$$\|\mathbf{c}\|_2^2 = \left\| \begin{bmatrix} \Re(\mathbf{c}) \\ \Im(\mathbf{c}) \end{bmatrix} \right\|_2^2 \tag{10}$$

for any complex vector $\mathbf{c}$; therefore Equations (7a) and (7b) can be used as they are, and Eq. (7c) corresponds only to addition of $\mathbf{u}$ and $\mathbf{r}$, which can be equivalently performed either in $\mathbb{C}^P$ or separately with the real and imaginary parts.

2 Sparse Audio Declipper (SPADE) – Derivation using ADMM

SPADE algorithm(s) approximate the solution of the following non-convex, NP-hard synthesis- or analysis-regularized inverse problems [2]:

$$\min_{\mathbf{x}, \mathbf{z}} \|\mathbf{z}\|_0 \quad \text{s.t.} \quad \mathbf{x} \in \Gamma(\mathbf{y}) \text{ and } \|A\mathbf{x} - \mathbf{z}\|_2 \le \epsilon \ \dots \ \text{A-SPADE}, \tag{11}$$

$$\min_{\mathbf{x},\mathbf{z}} \|\mathbf{z}\|_0 \quad \text{s.t.} \quad \mathbf{x} \in \Gamma(\mathbf{y}) \text{ and } \|\mathbf{x} - D\mathbf{z}\|_2 \leq \epsilon \ \dots \ \text{S-SPADE}, \tag{12}$$

where $\Gamma = \Gamma(\mathbf{y})$ is the set of feasible solutions, $\mathbf{x} \in \mathbb{R}^N$ is signal in the time domain, $\mathbf{z} \in \mathbb{C}^P$ are signal coefficients. Linear operator $D : \mathbb{C}^P \to \mathbb{R}^N$ is the synthesis operator and $A : \mathbb{R}^N \to \mathbb{C}^P$ is the analysis operator, and it holds $D = A^*$.

Because of the computational reasons, we will restrict ourself exclusively to Parseval tight frames, i.e. it holds

$$DD^* = A^*A = Id. \tag{13}$$

For such frames, it holds

$$\|A^*\mathbf{c}\|_2 \leq \|\mathbf{c}\|_2 \quad \text{while} \quad \|A^*\mathbf{c}\|_2 = \|\mathbf{c}\|_2 \text{ for } \mathbf{c} \in \mathcal{R}(A), \tag{14}$$

where $\mathcal{R}(A)$ denotes the range space of A.

Problems (12) and (11) can also be written as a sum of two indicator functions, such that

$$\min_{\mathbf{x},\mathbf{z},k} \iota_{\Gamma(\mathbf{y})}(\mathbf{x}) + \iota_{\ell_0 \leq k}(\mathbf{z}) \quad \text{s.t.} \quad \begin{array}{l} A\mathbf{x} - \mathbf{z} = 0 \ \dots \ \text{A-SPADE} \\ \mathbf{x} - D\mathbf{z} = 0 \ \dots \ \text{S-SPADE}, \end{array} \tag{15}$$

where $\iota_{\Gamma(\mathbf{y})}(\mathbf{x})$ is indicator function forcing the result to lie in the set of feasible solutions Γ and $\iota_{\ell_0 \leq k}(\mathbf{z})$ is a symbolic notation for indicator function $\iota_{\{\mathbf{z} \ | \ \|\mathbf{z}\|_0 \leq k\}}(\mathbf{z})$, that enforces the sparsity of the signal.

In the particular case of A-SPADE, our goal is to minimize the following

$$\operatorname*{argmin}_{\mathbf{x},\mathbf{z},k} \underbrace{\iota_{\Gamma(\mathbf{y})}(\mathbf{x})}_{f(\mathbf{x})} + \underbrace{\iota_{\ell_0 \leq k}(\mathbf{z})}_{g(\mathbf{z})} \quad \text{s.t.} \quad A\mathbf{x} - \mathbf{z} = 0, \tag{16}$$

which for fixed k correspond to problems of the form (2). However, the difference is the non-convexity of g, therefore we aim only at an *approximation* of (16) using ADMM [1]. We form the Augmented Lagrangian

$$L_\rho(\mathbf{x},\mathbf{y},\mathbf{z}) = \iota_{\Gamma(\mathbf{y})}(\mathbf{x}) + \iota_{\ell_0 \leq k}(\mathbf{z}) + \mathbf{y}^\top(A\mathbf{x} - \mathbf{z}) + \frac{\rho}{2}\|A\mathbf{x} - \mathbf{z}\|_2^2. \tag{17}$$

Using the scaled form, the Augmented Lagrangian can be also written as

$$L_\rho(\mathbf{x},\mathbf{z},\mathbf{u}) = \iota_{\Gamma(\mathbf{y})}(\mathbf{x}) + \iota_{\ell_0 \leq k}(\mathbf{z}) + \frac{\rho}{2}\|A\mathbf{x} - \mathbf{z} + \mathbf{u}\|_2^2 - \frac{\rho}{2}\|\mathbf{u}\|_2^2. \tag{18}$$

According to definition of ADMM in (7), we can form the ADMM algorithm for problem (16) as follows:

$$\mathbf{x}^{(i+1)} = \operatorname*{argmin}_{\mathbf{x}} \left(\iota_{\Gamma(\mathbf{y})}(\mathbf{x}) + \frac{\rho}{2}\|A\mathbf{x} - \mathbf{z}^{(i)} + \mathbf{u}^{(i)}\|_2^2 \right) \tag{19a}$$

$$\mathbf{z}^{(i+1)} = \operatorname*{argmin}_{\mathbf{z}} \left(\iota_{\ell_0 \leq k}(\mathbf{z}) + \frac{\rho}{2}\|A\mathbf{x}^{(i+1)} - \mathbf{z} + \mathbf{u}^{(i)}\|_2^2 \right) \tag{19b}$$

$$\mathbf{u}^{(i+1)} = \mathbf{u}^{(i)} + A\mathbf{x}^{(i+1)} - \mathbf{z}^{(i+1)}. \tag{19c}$$

Note that thanks to the indicator functions, the penalty parameter ρ does not play any role in finding the argument of the minima, therefore can be omitted. ADMM steps (19) can be also written in the constrained form, such that we get

$$\mathbf{x}^{(i+1)} = \operatorname*{argmin}_{\mathbf{x}} \|A\mathbf{x} - \mathbf{z}^{(i)} + \mathbf{u}^{(i)}\|_2^2 \quad \text{s.t.} \quad \mathbf{x} \in \Gamma \tag{20a}$$

$$\mathbf{z}^{(i+1)} = \operatorname*{argmin}_{\mathbf{z}} \|A\mathbf{x}^{(i+1)} - \mathbf{z} + \mathbf{u}^{(i)}\|_2^2 \quad \text{s.t.} \quad \|\mathbf{z}\|_0 \leq k \tag{20b}$$

$$\mathbf{u}^{(i+1)} = \mathbf{u}^{(i)} + A\mathbf{x}^{(i+1)} - \mathbf{z}^{(i+1)}. \tag{20c}$$

3

The **x**-update, i.e. the steps (19a) and (20a), is an orthogonal projection onto the set of feasible solutions Γ. The **z**-update, steps (19b) and (20b), is solved (not approximated!) by hard-thresholding operator $\mathcal{H}_k$, which sets all but k largest components in the magnitude of the input vector to zero. This is implied by the property

$$\underset{\mathbf{z}\in\mathbb{C}^P,\ \|\mathbf{z}\|_0\leq k}{\operatorname{argmin}} \|\mathbf{z}-\mathbf{s}\|_2^2 = \mathcal{H}_k(\mathbf{s}) \tag{21}$$

for any fixed $\mathbf{s}\in\mathbb{C}^P$.

Remark 2. *It is straightforward to show the property* (21). *First, it is clear that the optimal solution $\mathbf{z}_{\text{opt}}$ has sparsity k, unless $\|\mathbf{s}\|_0 < k$ and then $\mathbf{z}_{opt} = \mathbf{s}$. Supposing that it is not the trivial case, we will divide $\|\mathbf{z}-\mathbf{s}\|_2^2$ into two parts—the sum of squares of the coefficients of $\mathbf{z}$ that are forced to be zero by the condition $\|\mathbf{z}\|_0 \leq k$, and the rest. Apparently, the first part is minimized by setting the $P-k$ smallest coefficients of $\mathbf{s}$ to zero. Furthermore, the second part is minimized by setting the remaining k coefficients equal to corresponding values of $\mathbf{s}$, leading us to the solution $\mathbf{z}_{\text{opt}} = \mathcal{H}_k(\mathbf{s})$.*

Remark 3. *Step* (20a) *is a projection that seeks for a signal-domain solution. This is not trivial, however, along with [2], it can be transposed to a more convenient form, which is*

$$\mathbf{x}^{(i+1)} = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{x}-A^*(\mathbf{z}^{(i)}-\mathbf{u}^{(i)})\|_2^2 \quad s.t. \quad \mathbf{x}\in\Gamma, \tag{20a'}$$

i.e. a projection onto Γ that is easy to implement. We will now prove the equivalence of (20a) *and* (20a').

Denote $\mathbf{s} = \mathbf{z}^{(i)} - \mathbf{u}^{(i)}$ and define its unique factorization

$$\mathbf{s} = \underbrace{A\xi}_{\in\mathcal{R}(A)} + \underbrace{\varepsilon.}_{\perp\mathcal{R}(A)} \tag{22}$$

Using this, Eq. (20a) *can be rewritten as*

$$\mathbf{x}^{(i+1)} = \underset{\mathbf{x}}{\operatorname{argmin}} \|A(\mathbf{x}-\xi)-\varepsilon\|_2^2 \quad s.t. \quad \mathbf{x}\in\Gamma. \tag{23}$$

Because clearly $A(\mathbf{x}-\xi)\in\mathcal{R}(A)$ and $\varepsilon\perp\mathcal{R}(A)$, we can use the Pythagorean theorem and write

$$\mathbf{x}^{(i+1)} = \underset{\mathbf{x}}{\operatorname{argmin}} \|A(\mathbf{x}-\xi)\|_2^2 + \|\varepsilon\|_2^2 \quad s.t. \quad \mathbf{x}\in\Gamma. \tag{24}$$

Since we are searching for the argument of the minima, we can omit the term $\|\varepsilon\|_2^2$, which is independent of $\mathbf{x}$. Next, we will make use of the property of Parseval tight frames (14), *leading to the equivalence of Eq.* (24) *and*

$$\mathbf{x}^{(i+1)} = \underset{\mathbf{x}}{\operatorname{argmin}} \|\underbrace{A^*A}_{Id}(\mathbf{x}-\xi)\|_2^2 \quad s.t. \quad \mathbf{x}\in\Gamma. \tag{25}$$

Finally, we will show that $\xi = A^*\mathbf{s}$. Using $A\xi = \mathbf{s} - \varepsilon$ and the property (13), *we can write*

$$\xi = A^*(\mathbf{s}-\varepsilon) = A^*\mathbf{s} - A^*\varepsilon. \tag{26}$$

We know that $\varepsilon \perp \mathcal{R}(A) \Leftrightarrow \langle\varepsilon, A\omega\rangle = 0 \ \forall\omega$. Using the definition of adjoint operator, $\langle\varepsilon, A\omega\rangle = \langle A^*\varepsilon, \omega\rangle \ \forall\varepsilon \ \forall\omega$, therefore $\langle A^*\varepsilon, \omega\rangle = 0 \ \forall\omega \Leftrightarrow A^*\varepsilon = 0$ and $\xi = A^*\mathbf{s}$.

This completes the proof, because then $\|\mathbf{x}-\xi\|_2^2 = \|\mathbf{x}-A^*\mathbf{s}\|_2^2 = \|\mathbf{x}-A^*(\mathbf{z}^{(i)}-\mathbf{u}^{(i)})\|_2^2$ and that leads to Eq. (20a').

The SPADE algorithms, as introduced in the original paper [2], are shown in Alg. 1 (A-SPADE) and Alg. 2 (S-SPADE), respectively.

Algorithm 1: A-SPADE	**Algorithm 2:** S-SPADE
Require: $A, \mathbf{y}, M_{\mathrm{R}}, M_{\mathrm{H}}, M_{\mathrm{L}}, s, r, \epsilon$	**Require:** $D, \mathbf{y}, M_{\mathrm{R}}, M_{\mathrm{H}}, M_{\mathrm{L}}, s, r, \epsilon$
1 $\hat{\mathbf{x}}^{(0)} = \mathbf{y}, \mathbf{u}^{(0)} = \mathbf{0}, i = 1, k = s$	**1** $\hat{\mathbf{z}}^{(0)} = D^*\mathbf{y}, \mathbf{u}^{(0)} = \mathbf{0}, i = 1, k = s$
2 $\bar{\mathbf{z}}^{(i+1)} = \mathcal{H}_k\left(A\hat{\mathbf{x}}^{(i)} + \mathbf{u}^{(i)}\right)$	**2** $\bar{\mathbf{z}}^{(i+1)} = \mathcal{H}_k\left(\hat{\mathbf{z}}^{(i)} + \mathbf{u}^{(i)}\right)$
3 $\hat{\mathbf{x}}^{(i+1)} = \mathrm{argmin}_{\mathbf{x}} \\|A\mathbf{x} - \bar{\mathbf{z}}^{(i+1)} + \mathbf{u}^{(i)}\\|_2^2$ s.t. $\mathbf{x} \in \Gamma$	**3** $\hat{\mathbf{z}}^{(i+1)} = \mathrm{argmin}_{\mathbf{z}} \\|\mathbf{z} - \bar{\mathbf{z}}^{(i+1)} + \mathbf{u}^{(i)}\\|_2^2$ s.t. $D\mathbf{z} \in \Gamma$
4 if $\\|A\hat{\mathbf{x}}^{(i+1)} - \bar{\mathbf{z}}^{(i+1)}\\|_2 \leq \epsilon$ **then**	**4 if** $\\|\hat{\mathbf{z}}^{(i+1)} - \bar{\mathbf{z}}^{(i+1)}\\|_2 \leq \epsilon$ **then**
5 terminate	**5** terminate
6 else	**6 else**
7 $\mathbf{u}^{(i+1)} = \mathbf{u}^{(i)} + A\hat{\mathbf{x}}^{(i+1)} - \bar{\mathbf{z}}^{(i+1)}$	**7** $\mathbf{u}^{(i+1)} = \mathbf{u}^{(i)} + \hat{\mathbf{z}}^{(i+1)} - \bar{\mathbf{z}}^{(i+1)}$
8 $i \leftarrow i + 1$	**8** $i \leftarrow i + 1$
9 **if** $i \bmod r = 0$ **then**	**9** **if** $i \bmod r = 0$ **then**
10 $k \leftarrow k + s$	**10** $k \leftarrow k + s$
11 **end**	**11** **end**
12 go to 2	**12** go to 2
13 end	**13 end**
14 return $\hat{\mathbf{x}} = \hat{\mathbf{x}}^{(i+1)}$	**14 return** $\hat{\mathbf{x}} = D\hat{\mathbf{z}}^{(i+1)}$

While the paper [2] does not present the derivation of either A-SPADE or S-SPADE, we have shown in Sections 1 and 2 that A-SPADE perfectly fits the ADMM paradigm. On the other hand, the S-SPADE version in Alg. 2 follows a problem formulation which is different from the one the paper coped with. However, it is still derived from ADMM; it is easy to show that the problem formulation corresponding to the S-SPADE algorithm from [2] is

$$\min_{\mathbf{w},\mathbf{z}} \\|\mathbf{z}\\|_0 \quad \text{s.t.} \quad D\mathbf{w} \in \Gamma(\mathbf{y}) \text{ and } \\|\mathbf{w} - \mathbf{z}\\|_2 \leq \epsilon. \tag{27}$$

This original S-SPADE can be derived from (27) in the same way we described the A-SPADE derivation, starting from (11).

Section 3 introduces a new synthesis version of the SPADE algorithm derived from formulation (12), which is in some sense much more in line with the analysis approach than formulation (27) (therefore the "done right" in the title).

Remark 4. *Note that [2] reported that the projection step in S-SPADE is computationally expensive. Our recent paper [3] showed that there exists a one-step projection, making A-SPADE and S-SPADE identical from this point of view. It is a kind of paradox that we were able to speed up an algorithm that is derived in an inconsistent way (with the initial formulation), although producing reasonable results. In the new S-SPADE presented below, our fast projection step is not helpful.*

3   Synthesis Version of SPADE, Done Right

The problem formulation of this version of SPADE algorithm is very similar to (16), only with the difference that $\iota_{\ell_0 \leq k}(\mathbf{z})$ will be identified with $f(\mathbf{z})$ and likewise $\iota_{\Gamma(\mathbf{y})}(\mathbf{x})$ will be identified with $g(\mathbf{x})$. The primal variable, in this case, will be $\mathbf{z}$ representing signal coefficients, opposite to A-SPADE, where the primal variable was signal waveform, $\mathbf{x}$. We formulate the problem as

$$\underset{\mathbf{x},\mathbf{z},k}{\mathrm{argmin}} \underbrace{\iota_{\ell_0 \leq k}(\mathbf{z})}_{f(\mathbf{z})} + \underbrace{\iota_{\Gamma(\mathbf{y})}(\mathbf{x})}_{g(\mathbf{x})} \quad \text{s.t.} \quad D\mathbf{z} - \mathbf{x} = 0, \tag{28}$$

We again form the Augmented Lagrangian for this problem as:

$$L_\rho(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \iota_{\ell_0 \leq k}(\mathbf{z}) + \iota_{\Gamma(\mathbf{y})}(\mathbf{x}) + \mathbf{y}^\top(D\mathbf{z} - \mathbf{x}) + \frac{\rho}{2}\|D\mathbf{z} - \mathbf{x}\|_2^2. \tag{29}$$

As in (18), we use the scaled form

$$L_\rho(\mathbf{x}, \mathbf{z}, \mathbf{u}) = \iota_{\ell_0 \leq k}(\mathbf{z}) + \iota_{\Gamma(\mathbf{y})}(\mathbf{x}) + \frac{\rho}{2}\|D\mathbf{z} - \mathbf{x} + \mathbf{u}\|_2^2 - \frac{\rho}{2}\|\mathbf{u}\|_2^2, \tag{30}$$

leading to the ADMM steps as follows:

$$\mathbf{z}^{(i+1)} = \underset{\mathbf{z}}{\operatorname{argmin}} \|D\mathbf{z} - \mathbf{x}^{(i)} + \mathbf{u}^{(i)}\|_2^2 \quad \text{s.t.} \quad \|\mathbf{z}\|_0 \leq k \tag{31a}$$

$$\mathbf{x}^{(i+1)} = \underset{\mathbf{x}}{\operatorname{argmin}} \|D\mathbf{z}^{(i+1)} - \mathbf{x} + \mathbf{u}^{(i)}\|_2^2 \quad \text{s.t.} \quad \mathbf{x} \in \Gamma \tag{31b}$$

$$\mathbf{u}^{(i+1)} = \mathbf{u}^{(i)} + D\mathbf{z}^{(i+1)} - \mathbf{x}^{(i+1)}. \tag{31c}$$

When comparing ADMM steps of A-SPADE and S-SPADE, one can notice a different order of the minimization steps. This is caused by setting the variable $\mathbf{z}$ as the primal variable in S-SPADE. According to [1], it should be possible to freely choose the order of the f- and g-update steps. As a consequence, it is possible to first apply the hard thresholding and then the projection onto Γ in both A-SPADE and S-SPADE, being aware that the statement from [1] generally holds true for convex problems only.

The f-update (31a) is a challenging task to solve, but we rely on the frequent ADMM behaviour that it still converges even when the individual steps are computed (in contrast to A-SPADE) only approximately. We interpret Eq. (31a) as follows: we search for a k-sparse vector of coefficients $\mathbf{z}^{(i+1)}$, such that the synthesis operator D applied to these coefficients gives signal closest to $(\mathbf{x}^{(i)} - \mathbf{u}^{(i)})$. We suggest approximating the solution of this optimization task with the hard thresholding operator $\mathcal{H}_k$ applied to the analysis coefficients of $(\mathbf{x}^{(i)} - \mathbf{u}^{(i)})$, i.e.

$$\mathbf{z}^{(i+1)} \approx \mathbf{z}_{\text{appr}}^{(i+1)} = \mathcal{H}_k\left(D^*(\mathbf{x}^{(i)} - \mathbf{u}^{(i)})\right), \tag{32}$$

and we will show that this approximation is sufficiently close to the proper solution of (31a). Using the properties of Parseval tight frame (13) and (14), we can write

$$\left\|D\mathbf{z}^{(i+1)} - \mathbf{x}^{(i)} + \mathbf{u}^{(i)}\right\|_2^2 = \left\|D\mathbf{z}^{(i+1)} - DD^*(\mathbf{x}^{(i)} - \mathbf{u}^{(i)})\right\|_2^2 \tag{33a}$$

$$= \left\|D(\mathbf{z}^{(i+1)} - D^*(\mathbf{x}^{(i)} - \mathbf{u}^{(i)}))\right\|_2^2 \tag{33b}$$

$$\leq \left\|\mathbf{z}^{(i+1)} - D^*(\mathbf{x}^{(i)} - \mathbf{u}^{(i)})\right\|_2^2. \tag{33c}$$

Using the property (21), the norm (33c) is minimized by $\mathbf{z}_{\text{appr}}^{(i+1)}$ defined by (32). Therefore the error of the approximation in the time domain (i.e. the original norm in (33a)) is bounded by the minimal value of (33c) and thus we can expect it to be sufficiently small.

The solution to (31b) is obtained by a simple projection in the time domain.

The final version of the S-SPADE done right is shown in Algorithm 3.

Note that the computational complexity is identical in all three SPADEs and it is dominated by the cost of the transforms.

Remark 5. *Note that the ADMM step* (31a) *can be also approximated using other means, for example by a greedy algorithm such as the OMP, where the signal to be sparsely approximated is* $(\mathbf{x}^{(i)} - \mathbf{u}^{(i)})$*, the dictionary is D and the algorithm is forced to terminate after k iterations.*

Algorithm 3: S-SPADE done right

Require: $D, \mathbf{y}, M_{\mathrm{R}}, M_{\mathrm{H}}, M_{\mathrm{L}}, s, r, \epsilon$

1 $\hat{\mathbf{x}}^{(0)} = D^*\mathbf{y}, \mathbf{u}^{(0)} = \mathbf{0}, i = 1, k = s$

2 $\bar{\mathbf{z}}^{(i+1)} = \mathcal{H}_k\left(D^*(\hat{\mathbf{x}}^{(i)} - \mathbf{u}^{(i)})\right)$

3 $\hat{\mathbf{x}}^{(i+1)} = \mathrm{argmin}_{\mathbf{x}} \|D\bar{\mathbf{z}}^{(i+1)} - \mathbf{x} + \mathbf{u}^{(i)}\|_2^2 \;\; \text{s.t.} \;\; \mathbf{x} \in \Gamma$

4 **if** $\|D\bar{\mathbf{z}}^{(i+1)} - \hat{\mathbf{x}}^{(i+1)}\|_2 \leq \epsilon$ **then**

5 $\quad$ terminate

6 **else**

7 $\quad$ $\mathbf{u}^{(i+1)} = \mathbf{u}^{(i)} + D\bar{\mathbf{z}}^{(i+1)} - \hat{\mathbf{x}}^{(i+1)}$

8 $\quad$ $i \leftarrow i + 1$

9 $\quad$ **if** $i \bmod r = 0$ **then**

10 $\quad\quad$ $k \leftarrow k + s$

11 $\quad$ **end**

12 $\quad$ go to 2

13 **end**

14 **return** $\hat{\mathbf{x}} = \hat{\mathbf{x}}^{(i+1)}$
---

4 Equivalence of SPADEs in case of unitary operators

Note also that the three presented SPADEs are equivalent for unitary operators $D = A^* = A^{-1}$. This fact can be shown as an equivalence of formulations of the three problems—the resulting algorithms are equivalent as a consequence, since the derivation process is the same for all three. First, let us repeat the formulation (15) for A-SPADE and S-SPADE done right

$$\min_{\mathbf{x}, \mathbf{z}, k} \iota_{\Gamma(\mathbf{y})}(\mathbf{x}) + \iota_{\ell_0 \leq k}(\mathbf{z}) \quad \text{s.t.} \quad \begin{array}{l} A\mathbf{x} - \mathbf{z} = 0 \;\; \dots \;\; \text{A-SPADE} \\ \mathbf{x} - D\mathbf{z} = 0 \;\; \dots \;\; \text{S-SPADE,} \end{array} \tag{15}$$

and let (27) be reformulated in similar form as

$$\min_{\mathbf{w}, \mathbf{z}, k} \iota_{\Gamma(\mathbf{y})}(D\mathbf{w}) + \iota_{\ell_0 \leq k}(\mathbf{z}) \quad \text{s.t.} \quad \mathbf{w} - \mathbf{z} = 0. \tag{34}$$

The two formulations in (15) differ only by the constraint binding $\mathbf{x}$ with $\mathbf{z}$ and in the unitary case, one can be easily translated into the other by applying $D = A^{-1}$ (A-SPADE to S-SPADE) or $A = D^{-1}$ (S-SPADE to A-SPADE) on both sides of the constraint. To show the equivalence of (34) and (15), we substitute $\mathbf{x} = D\mathbf{w}$, leading to $\mathbf{w} = D^{-1}\mathbf{x} = A\mathbf{x}$. Formulation (34) then attains the form

$$\min_{\mathbf{x}, \mathbf{z}, k} \iota_{\Gamma(\mathbf{y})}(\underbrace{DA}_{Id}\mathbf{x}) + \iota_{\ell_0 \leq k}(\mathbf{z}) \quad \text{s.t.} \quad A\mathbf{x} - \mathbf{z} = 0, \tag{35}$$

which is the A-SPADE formulation from (15).

References

[1] S. P. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers." *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011. [Online]. Available: http://dblp.uni-trier.de/db/journals/ftml/ftml3.html#BoydPCPE11

[2] S. Kitić, N. Bertin, and R. Gribonval, "Sparsity and cosparsity for audio declipping: a flexible non-convex approach," in *LVA/ICA 2015 – The 12th International Conference on Latent Variable Analysis and Signal Separation*, Liberec, Czech Republic, Aug. 2015.

[3] P. Záviška, P. Rajmic, Z. Průša, and V. Veselý, "Revisiting synthesis model in sparse audio declipper," in *Latent Variable Analysis and Signal Separation*, Y. Deville, S. Gannot, R. Mason, M. D. Plumbley, and D. Ward, Eds. Cham: Springer International Publishing, 2018, pp. 429–445.