Joint Sparse Coding and Frame Optimization

Geoff Goehle and Benjamin Cowen, *Member, IEEE*

***Abstract*—We present two extensions of the Split Augmented Lagrangian Shrinkage Algorithm (SALSA), each addressing the joint optimization of a parameterized tight frame and their corresponding sparse coefficients. The two extensions, Parameter ADMM and Unrolled SALSA, can be adapted for either Basis Pursuit (constrained) or Basis Pursuit Denoising (unconstrained). The algorithms showcase the flexibility of the recently proposed Enveloped Sinusoid Parseval (ESP) Frames, and in particular, their aptitude for morphological component analysis (a sparsity-based approach to source separation) even when the source sparsity models have unknown parameters. We present a sample application using ESP frames to disentangle an additive mixture of two signal components, and empirically evaluate each algorithm's accuracy in recovering the components' parameters, their robustness in the presence of Gaussian white noise, and compare their performance to a traditional nonlinear least squares approach. The results indicate that Parameter ADMM and Unrolled SALSA outperform nonlinear least squares in low noise settings.**

***Index Terms*—Signal processing, optimization, sparse representations, compressed-sensing.**

I. Introduction

Sparse Coding is the problem of finding an efficient representation of some data in the range of a sparsifying transform, i.e., where the corresponding vectors of coefficients are mostly zero. When the sparsifying transform is known, the problem of recovering these coefficients from a data vector is a convex optimization problem that can be addressed by a wide variety of algorithms with mathematically guaranteed convergence [1], [2], notably including the Split Augmented Lagrangian Shrinkage Algorithm (SALSA) [3], [4]. However, when the sparsifying transform is also unknown, its joint optimization with the sparse coefficients is generally nonconvex [5], [6], [7]. This problem has been studied extensively for specific classes of signals [8], [9], [10] and is also known as dictionary learning [11].

Morphological Component Analysis is a generalization of Sparse Coding that seeks to separate additive mixtures by specifying their constituent components with differing sparse transforms [12], [13], [14]. The code inference subproblem is still convex, and can be solved by alternating direction method of multiplier (ADMM) based methods like SALSA; however, the general multi-dictionary learning problem is very ill-posed as the sparsifying transforms must efficiently represent their constituent component, but not the others [15], [16]. In practice, this requirement makes it difficult to provide hand-picked dictionaries for source separation [17], [18]. Enveloped Sinusoid Parseval (ESP) frames [19], [20] are a recent approach for generating sets of well-behaved sparse transforms from parameterized templates whose parameters, as shown in this work, are particularly amenable to optimization.

Many ℓ_1-based dictionary learning frameworks employ iterative optimization algorithms. A recently emerging technique in this field is to implement a truncated version of such an algorithm, i.e. for a fixed number of iterations, using a differentiable model [15], [21], [22]. This approach, nicknamed *unrolling*, uses error backpropagation to modify model parameters, commonly including dictionary-like components [23] and sometimes explicitly used for dictionary learning [24].

This letter presents two dictionary learning algorithms for the joint optimization of sparse transforms and their codes. First, a familiar ADMM-based approach to the problem is presented along with practical notes for accelerating convergence. Then, Unrolled SALSA is introduced, which uses an innovative forward-derivative approach to allow for arbitrarily large numbers of iterations (20,000 in the application presented herein).

Section II briefly establishes background material while the algorithms are described in Sections III and IV. Section V introduces a particular set of parameterized frames which are used in Section VI to demonstrate both joint optimization algorithms, analyze their robustness to noise, and compare their performance against a nonlinear least squares approach.

II. SALSA

Given a parameter vector $\boldsymbol{\theta}$ let $\mathbf{A}(\boldsymbol{\theta})$ be a tight frame [25] such that $\| \mathbf{A}(\boldsymbol{\theta})^* \mathbf{y} \|_2 = \alpha \| \mathbf{y} \|_2$ for all N-dimensional data vectors $\mathbf{y}$. Suppose also that $\mathbf{A}(\boldsymbol{\theta})$ is continuously differentiable as a function of $\boldsymbol{\theta}$. We are interested in solving the ℓ_1-regularization problems

$$(\boldsymbol{\theta}^*, \mathbf{x}^*) = \underset{\boldsymbol{\theta},\mathbf{x}}{\arg\min} \| \boldsymbol{\lambda} \odot \mathbf{x} \|_1 \text{ s.t. } \mathbf{A}(\boldsymbol{\theta})\, \mathbf{x} = \mathbf{y}, \tag{1}$$

$$(\boldsymbol{\theta}^*, \mathbf{x}^*) = \underset{\boldsymbol{\theta},\mathbf{x}}{\arg\min} \| \boldsymbol{\lambda} \odot \mathbf{x} \|_1 + \frac{1}{2} \| \mathbf{A}(\boldsymbol{\theta})\, \mathbf{x} - \mathbf{y} \|_2^2, \tag{2}$$

where $\boldsymbol{\lambda}$ is a weighting vector that controls both the degree to which sparsity is prioritized over reconstruction fidelity in (2) as well as relative prioritization between frame vectors. Given a fixed $\boldsymbol{\theta}$, (1) and (2) are well understood convex optimization problems, known as "Basis Pursuit" and "Basis Pursuit Denoising," respectively, both of which have guaranteed unique solutions that can be approximated using SALSA, which reframes

Manuscript received 15 December 2023; accepted 2 January 2024. Date of publication 8 January 2024; date of current version 17 January 2024. This work was sponsored in part by the Department of the Navy, Office of Naval Research under ONR under Award N00014-22-1-2620. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Giuseppe Thadeu Freitas de Abreu. *(Corresponding author: Geoff Goehle.)*

The authors are with the Pennsylvania State University Applied Research Laboratory, State College, PA 16803 USA (e-mail: goehle@psu.edu; ben.cowen@psu.edu).

Digital Object Identifier 10.1109/LSP.2024.3350810

Algorithm 1: Parameter ADMM.

```
if performing BP then $\beta = 1/\alpha$ else $\beta = (\mu + \alpha)^{-1}$
Initialize $\mu > 0$, $\gamma > 0$, $\boldsymbol{\theta}$, $\mathbf{x} = \mathbf{A}\,\mathbf{y}$, and $\mathbf{d} = 0$
while stopping criteria not satisfied do
  $\mathbf{v} \leftarrow \text{soft}(\mathbf{x} + \mathbf{d}, \boldsymbol{\lambda}\,/\mu) - \mathbf{d}$
  $\theta_i \leftarrow \theta_i - \gamma(\partial_{\theta_i}\,\mathbf{A}(\boldsymbol{\theta})\,\mathbf{v})^*(\mathbf{y} - \mathbf{A}(\boldsymbol{\theta})\,\mathbf{v})$
  $\mathbf{x} \leftarrow \mathbf{v} + \beta\,\mathbf{A}(\boldsymbol{\theta})^*(\mathbf{y} - \mathbf{A}(\boldsymbol{\theta})\,\mathbf{v})$
  $\mathbf{d} \leftarrow \mathbf{x} - \mathbf{v}$
end while
```

each problem using the Augmented Lagrangian Method [2] and solves the result using ADMM [26]. For (2), the augmented Lagrangian problem becomes

$$\mathbf{u} \leftarrow \underset{\mathbf{u}}{\arg\min}\, \|\,\boldsymbol{\lambda} \odot \mathbf{u}\,\|_1 + \frac{\mu}{2}\|\,\mathbf{u} - \mathbf{x} - \mathbf{d}\,\|_2^2 \tag{3}$$

$$\mathbf{x} \leftarrow \underset{\mathbf{x}}{\arg\min}\, \frac{1}{2}\|\,\mathbf{y} - \mathbf{A}\,\mathbf{x}\,\|_2^2 + \frac{\mu}{2}\|\,\mathbf{u} - \mathbf{x} - \mathbf{d}\,\|_2^2 \tag{4}$$

$$\mathbf{d} \leftarrow \mathbf{d} - (\mathbf{u} - \mathbf{x}) \tag{5}$$

for some $\mu > 0$ that affects convergence rate, but not the solution. Note that for tight frames (3) and (4) have exact solutions:

$$\mathbf{u} = \text{soft}\,(\mathbf{x} + \mathbf{d}, \boldsymbol{\lambda}\,/\mu) \tag{6}$$

$$\mathbf{x} = \mathbf{v} + \beta\,\mathbf{A}^*\,(\mathbf{y} - \mathbf{A}\,\mathbf{v}) \tag{7}$$

where $\mathbf{v} = \mathbf{u} - \mathbf{d}$, $\beta = (\mu + \alpha)^{-1}$, and soft is the soft thresholding function. The Basis Pursuit formulation has the same set of update formulas except with $\beta = 1/\alpha$.

Our goal is to generalize SALSA so that the optimization also includes the frame parameters $\boldsymbol{\theta}$.

III. Parameter ADMM

The Parameter ADMM algorithm extends the $\mathbf{x}$-subproblem (4) to

$$(\mathbf{x}, \boldsymbol{\theta}) \leftarrow \underset{\mathbf{x},\boldsymbol{\theta}}{\arg\min}\, \frac{1}{2}\|\,\mathbf{y} - \mathbf{A}(\boldsymbol{\theta})\,\mathbf{x}\,\|_2^2 + \frac{\mu}{2}\|\,\mathbf{u} - \mathbf{x} - \mathbf{d}\,\|_2^2. \tag{8}$$

We observe that when this is addressed with alternating minimization [15], the iteration can be collapsed into a single step by plugging in the closed-form solution (7) to the $\boldsymbol{\theta}$-subproblem. After some manipulation, we have

$$\boldsymbol{\theta} \leftarrow \underset{\boldsymbol{\theta}}{\arg\min}\, c\|\,\mathbf{A}(\boldsymbol{\theta})\,\mathbf{v} - \mathbf{y}\,\|_2^2 \tag{9}$$

where c is a constant depending on μ and β. Problem (9) is generally non-convex, but for amenable parameterizations it is continuously differentiable. We opt for a first-order approach,

$$\theta_i \leftarrow \theta_i - \gamma(\partial_{\theta_i}\,\mathbf{A}(\boldsymbol{\theta})\,\mathbf{v})^*\,(\mathbf{y} - \mathbf{A}(\boldsymbol{\theta})\,\mathbf{v})\,, \tag{10}$$

where γ is our learning rate. The full procedure is described in Algorithm 1.

In practice we modify Algorithm 1 in two ways: first, $\boldsymbol{\theta}$ is updated slowly, only once for every M updates of $\mathbf{v}$ and $\mathbf{x}$. This allows $\mathbf{v}$ and $\mathbf{x}$ to gather momentum between updates. When M is large enough that they converge, we have recovered Alternating Minimization, running SALSA between updates of $\boldsymbol{\theta}$. We heuristically selected $M = 20$ for results herein, as it produces sufficient SALSA convergence to support stable updating of $\boldsymbol{\theta}$. Second, instead of a constant learning rate γ, an adaptive learning rate is used, which allows $\boldsymbol{\theta}$ to carry momentum from past updates. We utilize the common Adaptive Moment Estimation (Adam) algorithm [27] to vary the learning rate.

Algorithm 2: Unrolled SALSA.

```
if performing BP then $\beta = 1/\alpha$ else $\beta = (\mu + \alpha)^{-1}$
Initialize $\mu > 0$, $\gamma > 0$, $\boldsymbol{\theta}$, $\mathbf{x} = \mathbf{A}\,\mathbf{y}$, and $\mathbf{d} = 0$
while stopping criteria not satisfied do
  $\mathbf{v} \leftarrow \text{soft}(\mathbf{x} + \mathbf{d}, \boldsymbol{\lambda}\,/\mu) - \mathbf{d}$
  $\partial_{\theta_i}\,\mathbf{v} \leftarrow \partial_{\theta_i}\text{soft}(\mathbf{x} + \mathbf{d}, \boldsymbol{\lambda}\,/\mu) \odot (\partial_{\theta_i}\,\mathbf{x} + \partial_{\theta_i}\,\mathbf{d}) - \partial_{\theta_i}\,\mathbf{d}$
  $\mathbf{x} \leftarrow \mathbf{v} + \beta\,\mathbf{A}(\boldsymbol{\theta})^*(\mathbf{y} - \mathbf{A}(\boldsymbol{\theta})\,\mathbf{v})$
  $\partial_{\theta_i}\,\mathbf{x} \leftarrow \partial_{\theta_i}\,\mathbf{v} + \beta\partial_{\theta_i}\,\mathbf{A}(\boldsymbol{\theta})^*(\mathbf{y} - \mathbf{A}(\boldsymbol{\theta})\,\mathbf{v})$
      $-\beta\,\mathbf{A}(\boldsymbol{\theta})^*(\partial_{\theta_i}\,\mathbf{A}(\boldsymbol{\theta})\,\mathbf{v} + \mathbf{A}(\boldsymbol{\theta})\partial_{\theta_i}\,\mathbf{v})$
  $\mathbf{d} \leftarrow \mathbf{x} - \mathbf{v}$
  $\partial_{\theta_i}\,\mathbf{d} \leftarrow \partial_{\theta_i}\,\mathbf{x} - \partial_{\theta_i}\,\mathbf{v}$
  if performing BP then
    $\theta_i \leftarrow \theta_i - \gamma\sum_j \lambda[j]|x[j]|\,\text{Re}(\frac{\partial_{\theta_i}x[j]}{x[j]})$
  else
    $\theta_i \leftarrow \theta_i - \gamma\Big(\sum_j \lambda[j]|x[j]|\,\text{Re}(\frac{\partial_{\theta_i}x[j]}{x[j]})$
        $+(\mathbf{A}(\boldsymbol{\theta})\,\mathbf{x} - \mathbf{y})^*(\partial_{\theta_i}\,\mathbf{A}(\boldsymbol{\theta})\,\mathbf{x} + \mathbf{A}(\boldsymbol{\theta})\partial_{\theta_i}\,\mathbf{x})\Big)$
  end if
end while
```

IV. Unrolled SALSA

Another way to address Problems (1) and (2) is to treat SALSA (3)–(5) as a differentiable function, using error backpropagation to apply gradient-based updates to $\boldsymbol{\theta}$. This approach is known as unrolling [15], [21]. However, to implement backpropagation on unrolled algorithms, computer memory must be allocated to save the intermediate outputs of each iteration, precluding the use of many iterations.

We propose to instead compute the *forward* derivative of SALSA during inference, which supports an arbitrary number of iterations. We use Wirtinger derivatives [28] to account for vector spaces over $\mathbb{C}$, which admit convenient formulations for the complex soft function and ℓ_1-norm cost functions:

$$\partial_\tau|x| = \begin{cases} |x|\,\text{Re}\left[\frac{\partial_\tau x}{x}\right] & x \neq 0 \\ 0 & x = 0 \end{cases},$$

$$\partial_\tau\text{soft}(x, T) = \begin{cases} \left(1 - \frac{T}{2}\frac{1}{|x|}\right)\partial_\tau x + \frac{T}{2}\frac{x(\partial_\tau x)^*}{|x|x^*} & |x| > T \\ 0 & |x| \leq T. \end{cases}$$

From these, the derivatives of the cost functions (1)–(2) and their solutions via SALSA (3)–(5) follow, suggesting a gradient descent update formula for $\boldsymbol{\theta}$. This procedure, Unrolled SALSA, is presented in Algorithm 2.

Note that the computation of the forward derivative couples the variables' update formulas, making the order of the steps more important than in SALSA or ADMM. As with Parameter ADMM, we suggest that $\boldsymbol{\theta}$ be updated only once every M updates of $\mathbf{v}$ and $\mathbf{x}$, where M is chosen so that there is sufficient SALSA convergence to support alternating minimization. We use $M = 20$ for the application presented below. Furthermore, we again use the Adam optimizer [27] to adaptively adjust the learning rate γ. Note that with this approach, after each $\boldsymbol{\theta}$ update, we effectively hot-start the next round of SALSA computations with the coefficients *and* the derivatives from the previous round of SALSA updates.

This last point is particularly subtle. The SALSA subproblem is guaranteed to converge to a global minimum, and tends to converge faster given hot-started coefficients. The computation of the gradient, however, accumulates error since the hot-started derivatives are only approximately correct given the updated parameters. This accumulation of error thus requires the derivative estimate to be periodically reset to zero to prevent numerical instability. A formal analysis of this algorithmic mechanism is outside the scope of this work.

V. Enveloped Sinusoid Parseval Frames

While Algorithms 1 and 2 both work for any differentiably parameterized tight frame, we use ESP frames [19], [20]: Parseval frames defined by a set of L complex envelopes $\{\mathbf{e}_l\}$ modulated by discrete Fourier Transform basis functions and circularly shifted in time. Frame coefficients are indexed by envelope number l, frequency bin k, and time shift index m. The analysis matrix is defined to be

$$A[l,k,m,n] = \frac{e_l[m - n \bmod N]}{\sqrt{NL}\| \, \mathbf{e}_l \, \|_2} \exp\left(\frac{2\pi i k(n-m)}{N}\right), \tag{11}$$

where l, k, m are jointly represented as the single index j for the purposes of Algorithms 1 and 2. Theorem 1 of [19] proves that any collection of envelopes, including any parameterized collection $\mathbf{e}_l(\boldsymbol{\theta})$, form a tight-frame in the ESP framework. The optional normalization term in (11) further guarantees that $\mathbf{A}(\boldsymbol{\theta})$ is a Parseval frame with $\alpha = 1$ for *any* specific values of $\boldsymbol{\theta}$ (at the cost of a much more complicated derivative).

ESP frames are specifically useful in this context for two reasons. First, because any set of envelopes can define an ESP frame, we observe that the derivative of an ESP frame with respect to parameter θ_i is another ESP frame defined by the envelopes $\frac{\partial \mathbf{e}_l}{\partial \theta_i}$. We call this the *derivative frame*, $\partial_{\theta_i} \mathbf{A}$, and note that it may not be a Parseval frame. Second, there are robust, GPU-accelerated, FFT-based formulas for computing the analysis and synthesis operators of ESP frames [20], which make it feasible for iterative algorithms.

VI. Application

In this section we demonstrate both algorithms with a parameter estimation problem. We will consider a noisy additive mixture of two circularly shifted and modulated envelopes e_l given by

$$e_1(t) = \exp\left(\frac{-0.5(t-t_1)^2}{10^{2\theta_1}}\right), \tag{12}$$

$$e_2(t) = \begin{cases} \exp\left(\frac{t-t_2}{10^{\theta_2}}\right) & t < t_2 \\ \exp\left(\frac{-(t-t_2)}{10^{\theta_3}}\right) & t > t_2. \end{cases} \tag{13}$$

We fix $t_1 = 2.5$ms and $t_2 = 1$ms throughout and view θ_i as our frame parameters. To illustrate the flexibility of Parameter ADMM and Unrolled SALSA, the envelope parameters are defined such that they control different aspects of two morphologically distinct envelopes and are not in one-to-one correspondence with the envelopes. They are all exponential scale factors with common units of $\log$ s, however, and will be grouped together for error analysis.

We define the continuous signal y by

$$y(t) = \sum_{l=1}^{2} a_l e_l(t - \tau_l) \exp(2\pi i f_l (t - \tau_l)) \tag{14}$$

with the following set of ground truth parameters:

$$\begin{array}{lll} \theta_1 = -3.0 & \theta_2 = -3.5 & \theta_3 = -4.0 \\ a_1 = 3.0 & \tau_1 = 0.7 \text{ ms} & f_1 = 11 \text{ kHz} \\ a_2 = 4.0 & \tau_2 = 0.2 \text{ ms} & f_2 = 7 \text{ kHz}. \end{array}$$

For the observation $\mathbf{y}$ we discretize $y(t)$ with $N = 1000$ samples at a sampling frequency of $f_s = 100$kHz.

We wish to recover all nine of the above parameters from noisy realizations of $\mathbf{y}$, utilizing Algorithms 1 and 2 with ESP frames $\mathbf{A}(\boldsymbol{\theta})$ generated from envelopes $e_l(\boldsymbol{\theta})$ defined by (discretizations of) (12) and (13). In addition to directly estimating θ_l, we estimate the time shift and modulation frequency parameters as the time shift and modulation frequency associated to the frame vector with the largest coefficient amplitude for each component.

We will use the Basis Pursuit Denoising variant of both techniques with cost function (2). Following [29], for a given signal realization $\mathbf{y}$ we let $\lambda_{\max}$ be the threshold above which the SALSA solution is zero. For each signal we will perform 20,000 iterations of Algorithms 1 and 2 with (scalar) $\lambda = 0.1\lambda_{\max}$ and $\boldsymbol{\theta}$-updates every 20 iterations using Adam-managed gradient descent. For Unrolled SALSA, the derivatives will be reset to zero every 10 $\boldsymbol{\theta}$-updates. These hyperparameter values were chosen heuristically for this demonstration and determining optimal values is a direction of future work.

As a point of comparison we also perform traditional Nonlinear Least Squares (LS) directly on (14) by using gradient descent techniques to minimize the ℓ_2-error between the noisy signal $\mathbf{y}$ and a clean version constructed using (14) and estimated parameters. Notably we specify the (correct) number of signal components as part of the signal model, which gives the nonlinear least squares approach a significant advantage over the ℓ_1-regularization based techniques.

For consistency, 1000 iterations of Adam-managed nonlinear least squares gradient descent will be performed for each signal realization. Additionally, while the same representation of the envelope parameters is used for all three optimization methods, for Nonlinear LS we normalize the amplitude, time shift and modulation frequency parameters by the maximum parameter value and apply a logit transformation. This improves the performance of the gradient descent by putting all nine parameters on a similar scale.

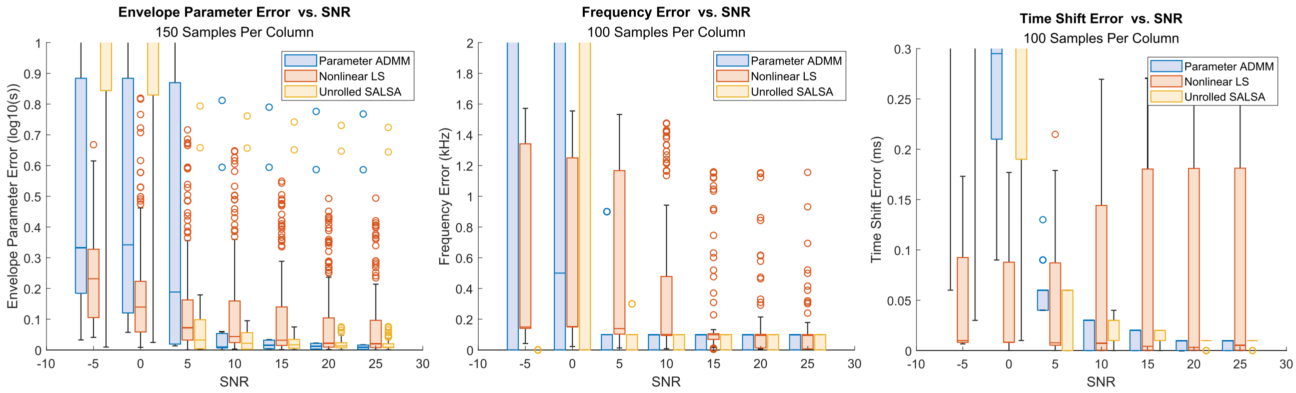

Fig. 1. Envelope parameter error (left), modulation frequency parameter error (middle) and time shift parameter error (right) vs. SNR for Parameter ADMM, Nonlinear LS and Unrolled SALSA. For each noise level, parameter estimates were generated using 50 signal realizations with normally distributed initial conditions. Not pictured are the full box plots for Parameter ADMM and Unrolled SALSA at SNR of −5 dB and 0 dB, which extend above the plotted range, and several (2–3) outliers for Parameter ADMM and Unrolled SALSA at SNR of 0 dB to 25 dB.

Each approach was applied to realizations of $\mathbf{y}$ with additive white Gaussian noise at SNR ranging from −5 dB to 25 dB. For each noise level 50 noise realizations were generated, and for each noise realization initial envelope parameters were sampled from Gaussian distributions centered on the true envelope parameters with a 0.25 standard deviation. In addition, initial amplitude, time shift, and modulation frequency parameters were generated for the Nonlinear LS approach. These initial values were also selected from Gaussian distributions centered on the true (normalized and logit-transformed) parameter values with a 0.05 standard deviation. This provides the LS estimation with a significant amount of prior information compared to the other two approaches. After initialization each method was applied as described above to produce the final parameter estimates.

On average Parameter ADMM training sessions took 21 s, while Unrolled SALSA sessions lasted an average of 71 s. For comparison Nonlinear LS training lasted only 5 s on average. Fig. 1 displays the resulting envelope parameter estimate error, modulation frequency estimate error, and time shift estimate error for each method as a function of SNR. Broadly speaking, Algorithms 1 and 2 performed similarly, with Parameter ADMM doing slightly better. For signals with positive SNR, Parameter ADMM and Unrolled SALSA do a good job of estimating the underlying signal parameters. The error in the time shift and modulation frequency parameters is generally below the resolution of our discretized signal. The envelope parameter error ranges from 0.01 to 0.02 $\log \text{s}$ (2 to 45μs, depending on the parameter value). Despite its advantages, the nonlinear least squares approximation performs more poorly than the ℓ_1-based approaches for high SNR signals, with a larger median error and more outliers.

The estimation fidelity for the sparsity-based approaches begins to drop off at 5 dB, with the envelope parameter error for Parameter ADMM performing particularly poorly. For signals of 0 dB SNR or less neither Parameter ADMM nor Unrolled SALSA effectively approximate any of the signal parameters. At these noise levels and λ-values there is insufficient sparsity to support parameter estimation. On the other hand, Nonlinear LS significantly outperforms Algorithms 1 and 2 for low SNR signals. This is not unexpected, since Nonlinear LS includes a signal model with the correct number of components and initial values for *all* parameters. Overall we find that when either Parameter ADMM or Unrolled SALSA is successful in achieving a sparse solution, they outperform a direct least squares approach for parameter estimation. However both methods suffer in the presence of high noise levels.

VII. Discussion

We have presented two extensions of SALSA that jointly optimize parameterized sparsifying transforms and the corresponding sparse coefficients. Both methods were tested using differentiably parameterized ESP frames with two morphologically distinct envelopes and three parameters. We find that Parameter ADMM and Unrolled SALSA are each robust to noise, assuming a positive SNR, with Parameter ADMM outperforming Unrolled SALSA in terms of speed and envelope parameter error and both methods outperforming a Nonlinear LS approach.

While Section VI indicates that Parameter ADMM outperforms Unrolled SALSA, these results are particular to the choice of frame and its parameterization. We have presented results for ESP frames, but both Algorithms 1 and 2 extend to any differentiably parameterized frame, which may exhibit different relative characteristics. At a high level, Parameter ADMM is faster and should be preferred when frame optimization is performed for a single signal. Unrolled SALSA can be used to derive a frame from a set of training signals which is optimized for a specific number of vanilla SALSA iterations. This is useful for building a frame which will then used for traditional MCA on a multitude of signals.

This has been a preliminary effort and there are many potential directions for future research. In addition to utilizing Parameter ADMM and Unrolled SALSA on a wider range of frames, and testing the denoising capabilities of the algorithms by optimizing λ for high noise levels, there is also the potential to combine the algorithms into a hybrid approach or to augment them with mode collapse prevention techniques. Application of these algorithms could be used to improve parameter estimation in signal analysis and adaptive sparse signal representation.

REFERENCES

[1] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Commun. Pure Appl. Math.: A. J. Issued Courant Inst. Math. Sci.*, vol. 57, no. 11, pp. 1413–1457, 2004.

[2] S. Boyd and L. Vandenberghe, *Convex Optimization.*, Cambridge, U.K.: Cambridge Univ. Press, 2004.

[3] M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo, "Fast image recovery using variable splitting and constrained optimization," *IEEE Trans. Image Process.*, vol. 19, no. 9, pp. 2345–2356, Sep. 2010.

[4] I. Selesnick, "L1-norm penalized least squares with SALSA," 2014. [Online]. Available: http://cnx.org/content/m48933/

[5] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 689–696.

[6] J. Shi, X. Ren, G. Dai, J. Wang, and Z. Zhang, "A non-convex relaxation approach to sparse dictionary learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 1809–1816.

[7] C. Bao, H. Ji, Y. Quan, and Z. Shen, "L0 norm based dictionary learning by proximal methods with global convergence," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 3858–3865.

[8] C. Herglotz, N. Genser, and A. Kaup, "Rate-distortion optimal transform coefficient selection for unoccupied regions in video-based point cloud compression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 11, pp. 7996–8009, Nov. 2022.

[9] L. Lovisolo, M. P. Tcheou, E. A. B. da Silva, M. A. M. Rodrigues, and P. S. R. Diniz, "Modeling of electric disturbance signals using damped sinusoids via atomic decompositions and its applications," *EURASIP J. Adv. Signal Process.*, vol. 2007, no. 1, Dec. 2007, Art. no. 029507.

[10] M. P. Tcheou, L. Lovisolo, E. A. B. da Silva, M. A. M. Rodrigues, and P. S. R. Diniz, "Optimum rate-distortion dictionary selection for compression of atomic decompositions of electric disturbance signals," *IEEE Signal Process. Lett.*, vol. 14, no. 2, pp. 81–84, Feb. 2007.

[11] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, 1996.

[12] M. Elad, J.-L. Starck, P. Querre, and D. L. Donoho, "Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA)," *Appl. Comput. Harmon. Anal.*, vol. 19, no. 3, pp. 340–358, 2005.

[13] J. L. Starck, M. Elad, and D. L. Donoho, "Image decomposition via the combination of sparse representations and a variational approach," *IEEE Trans. Image Process.*, vol. 14, no. 10, pp. 1570–1582, Oct. 2005.

[14] I. W. Selesnick, "A new sparsity-enabled signal separation method based on signal resonance," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2010, pp. 4150–4153.

[15] B. Cowen, A. N. Saridena, and A. Choromanska, "LSALSA: Accelerated source separation via learned sparse coding," *Mach. Learn.*, vol. 108, pp. 1307–1327, 2019.

[16] D. L. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization," *Proc. Nat. Acad. Sci.*, vol. 100, no. 5, pp. 2197–2202, 2003.

[17] B. Singh and H. Wagatsuma, "A removal of eye movement and blink artifacts from EEG data using morphological component analysis," *Comput. Math. Methods Med.*, vol. 2017, pp. 1–17, 2017.

[18] A. Parekh, I. W. Selesnick, D. M. Rapoport, and I. Ayappa, "Detection of K-complexes and sleep spindles (DETOKS) using sparse optimization," *J. Neurosci. Methods*, vol. 251, pp. 37–46, 2015.

[19] G. Goehle, B. Cowen, J. D. Park, and D. C. Brown, "Enveloped sinusoid parseval frames," in *Proc. OCEANS*, 2022, pp. 1–10.

[20] G. Goehle, B. Cowen, T. E. Blanford, J. Daniel Park, and D. C. Brown, "Approximate extraction of late-time returns via morphological component analysis," *J. Acoustical Soc. Amer.*, vol. 153, no. 5, 2023, Art. no. 2838.

[21] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proc. 27th Int. Conf. Int. Conf. Mach. Learn.*, 2010, pp. 399–406.

[22] T. Chen et al., "Learning to optimize: A primer and a benchmark," *J. Mach. Learn. Res.*, vol. 23, no. 1, pp. 8562–8620, 2022.

[23] J. Sun et al., "Deep ADMM-Net for compressive sensing MRI," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 10–18.

[24] N. Janjušević, A. Khalilian-Gourtani, and Y. Wang, "CDLNet: Noise-adaptive convolutional dictionary learning network for blind denoising and demosaicing," *IEEE Open J. Signal Process.*, vol. 3, pp. 196–211, 2022.

[25] D. Han, K. Kornelson, D. Larson, and E. Weber, *Frames for Undergraduates.*, Providence, Rhode Island: Amer. Math. Soc., 2007.

[26] T. Goldstein, B. O'Donoghue, S. Setzer, and R. Baraniuk, "Fast alternating direction optimization methods," *SIAM J. Imag. Sci.*, vol. 7, no. 3, pp. 1588–1623, 2014.

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, pp. 1–13, 2014.

[28] W. Wirtinger, "Zur formalen theorie der funktionen von mehr komplexen veränderlichen," *Mathematische Annalen*, vol. 97, no. 1, pp. 357–375, Dec. 1927, doi: 10.1007/bf01447872.

[29] A. Xenaki and Y. Pailhas, "Compressive synthetic aperture sonar imaging with distributed optimization," *J. Acoustical Soc. Amer.*, vol. 146, no. 3, pp. 1839–1850, 2019.