Learning Convex Regularizers for Optimal Bayesian Denoising

Ha Q. Nguyen, Emrah Bostan, *Member, IEEE*, and Michael Unser, *Fellow, IEEE*

***Abstract*—We propose a data-driven algorithm for the Bayesian estimation of stochastic processes from noisy observations. The primary statistical properties of the sought signal are specified by the penalty function (i.e., negative logarithm of the prior probability density function). Our alternating direction method of multipliers (ADMM) based approach translates the estimation task into successive applications of the proximal mapping of the penalty function. Capitalizing on this direct link, we define the proximal operator as a parametric spline curve and optimize the spline coefficients by minimizing the average reconstruction error for a given training set. The key aspects of our learning method are that the associated penalty function is constrained to be convex and the convergence of the ADMM iterations is proven. As a result of these theoretical guarantees, adaptation of the proposed framework to different levels of measurement noise is extremely simple and does not require any retraining. We apply our method to estimation of both sparse and nonsparse models of Lévy processes for which the minimum mean square error (MMSE) estimators are available. We carry out a single training session for a fixed level of noise and perform comparisons at various signal-to-noise ratio values. Simulations illustrate that the performance of our algorithm are practically identical to the one of the MMSE estimator irrespective of the noise power.**

***Index Terms*—Bayesian estimation, learning for inverse problems, alternating direction method of multipliers, convolutional neural networks, back propagation, sparsity, convex optimization, proximal methods, monotone operator theory.**

I. Introduction

STATISTICAL inference is a central theme in the theory of inverse problems. Bayesian methods, in particular, have been successfully used in several signal processing problems [1]–[3]. Among these is the estimation of signals under the additive white Gaussian noise (AWGN) hypothesis, which we shall consider throughout this paper. Specifically, we are interested in the problem of estimating a signal $\boldsymbol{x} \in \mathbb{R}^N$ from its noisy observation

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{n}, \tag{1}$$

where $\boldsymbol{n}$ is AWGN of variance σ^2. Conventionally, the unobservable signal $\boldsymbol{x}$ is modeled as a random object with a prior probability density function (pdf) p_X and the estimation is performed by assessing the posterior pdf $p_{X|Y}$ that characterizes the problem statistically. In addition to being important on its own right, this classical problem has recently gained a significant amount of interest. The main reason of the momentum is that Bayesian estimators can be directly integrated—as "denoisers"—into algorithms that are designed for more sophisticated inverse problems [4]. In plain terms, employing a more accurate denoising technique helps one improve the performance of the subsequent reconstruction method. Such ideas have been presented in various applications including deconvolution [5], super-resolved sensing [6], and compressive imaging [7], to name just a few.

The maximum a posteriori (MAP) inference is by far the most widely used Bayesian paradigm due its computational convenience [8]. This approach assumes the existence of a whitening operator $\boldsymbol{L}$ such that the pdf of the transformed signal $\boldsymbol{u} = \boldsymbol{L}\boldsymbol{x}$ is separable, which allows us to express the MAP estimation as

$$\hat{\boldsymbol{x}}_{\mathrm{MAP}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \left\{ \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{x}\|_2^2 + \sigma^2 \sum_{i=1}^{N} \Phi_U\left([\boldsymbol{L}\boldsymbol{x}]_i\right) \right\}, \tag{2}$$

where $\Phi_U = -\log p_U$ is called the *penalty function* and p_U is the pdf of each component of $\boldsymbol{u}$. Through this expression, compatibility of MAP with the regularized least-squares approach is well-understood. For example, by this parallelism, the implicit statistical links between the popular sparsity-based methods [9] and MAP considerations based on generalized Gaussian, Laplace, or hyper-Laplace priors is established [10]–[12]. Moreover, recent iterative optimization techniques including (fast) iterative shrinkage/thresholding algorithm ((F)ISTA) [13]–[15] and ADMM [16] allow us to handle the optimization problem (2) very efficiently.

Fundamentally, the estimation performance of MAP is differentiated by the underlying prior model. When the inherent nature of the underlying signal is (fully or partially) deterministic, identification of the correct prior is challenging. Fitting

Manuscript received May 15, 2017; revised September 26, 2017; accepted November 10, 2017. Date of publication November 24, 2017; date of current version January 16, 2018. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Namrata Vaswani. This work was supported by the Swiss National Science Foundation under Grant 200020-162343 and the European Research Council under Grant ERC-692726-GlobalBioIm. *(Corresponding author: Ha Q. Nguyen.)*

H. Q. Nguyen was with the Biomedical Imaging Group, École Polytechnique Fédérale de Lausanne, Lausanne CH-1015, Switzerland. He is now with the Viettel Research & Development Institute, Hanoi, Vietnam (e-mail: hanq8@viettel.com.vn).

E. Bostan was with the Biomedical Imaging Group, École Polytechnique Fédérale de Lausanne, Lausanne CH-1015, Switzerland. He is now with the Computational Imaging Lab, Department of Electrical Engineering and Computer Science, University of California, Berkeley, Berkeley, CA 94720 USA (e-mail: bostan@berkeley.edu).

M. Unser is with the Biomedical Imaging Group, École Polytechnique Fédérale de Lausanne, Lausanne CH-1015, Switzerland (e-mail: michael.unser@epfl.ch).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2017.2777407

1053-587X © 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

statistical models to such signals (or collections of them) is feasible [17]. Yet, the apparent downside is that the reference pdf, which specifies the inference, can be arbitrary. Even when the signal of interest is purely stochastic and the prior is exactly known, deviations from the initial statistical assumptions is observed [18]. More importantly, mathematical characterization of the estimation error of the MAP solution (that is the maximizer of the posterior pdf) by means of mean-square error (MSE) is available only in limited cases [19]. Hence, algorithms driven by rigorous MAP considerations can still be suboptimal with respect to MSE [20], [21]. These observations necessitate revisiting MAP-like formulations from the perspective of estimation accuracy instead of strict derivations based on the prior model.

A. Overview of Related Literature

Several works have aimed at improving the performance of MAP. Cho *et al.* have introduced a nonconvex method to enforce the strict fit between the signal (or its attributes) and their choice of prior distribution [22]. Gribonval has shown that the MMSE can actually be stated as a variational problem that is in spirit of MAP [23]. Based on the theory of continuous-domain sparse stochastic processes [24], Amini *et al.* have analyzed the conditions under which the performance of MAP can be MSE-optimal [25]. In [26], Bostan *et al.* have investigated the algorithmic implications of various prior models for the proximal (or the shrinkage) operator that takes part in the ADMM steps. Accordingly, Kazerouni *et al.* [27] and Tohidi *et al.* [28] have demonstrated that MMSE performance can be achieved for certain type of signals if the said proximal operator is replaced with carefully chosen MMSE-type shrinkage functions. Such methods, however, rely on the full knowledge of the prior model, which significantly limits their applicability.

Modification of the proximal operators have also been investigated based on deterministic principles. In particular, motivated by the outstanding success of convolutional neural networks (CNNs) [29], several researchers have used learning-based methods to identify model parameters (thus the proximal mapping). In this regard, Gregor and LeCun [30], and Kamilov and Mansour [31] have considered sparse encoding applications and replaced the soft-thresholding step in (F)ISTA with a learned proximal mapping. Schmidt and Roth [32] have proposed to learn different shrinkage functions for each iteration of the half-quadratic minimization method with applications in image restoration. Meanwhile, Chen *et al.* have developed a similar learning strategy for the gradient descent method [33], [34]. Yang *et al.* have applied learning to piecewise-linear proximal operators of ADMM, which also vary with iteration, for improved magnetic resonance (MR) image reconstruction [35]. A variant of these methods is considered by Lefkimmiatis [36]. More relevant to the present context, Samuel and Tappen have learned the model parameters of MAP estimators for continuous-valued Markov random fields (MRFs) [37]. What is common in all these techniques is that the proximal algorithm at hand is trained without any restrictions. This makes it hard to say anything about the signal reconstruction in the testing phase using the learned proximal operators. By contrast, we propose in this paper to learn a single proximal operator of some convex penalty function (or regularizer) for every iteration of ADMM, so that the iterative architecture of the reconstruction still stays within the realm of convex optimization.

B. Contributions

We revisit the MAP problem for signal denoising that is cast as the minimization of a quadratic fidelity term regularized by a penalty function. The latter captures the statistics of a collection of clean signals. The problem is solved via ADMM by iteratively applying the proximal operator associated with the penalty function. The main advantage of ADMM over other iterative methods such as ISTA/FISTA is that it can decouple the effects of the whitening operator $\boldsymbol{L}$ and of the scalar penalty function Φ_U in (2), which makes the learning of this function feasible even when $\boldsymbol{L}$ is not the identity operator. When the proximal operator in ADMM is replaced with a trainable (shrinkage) function, we call the reconstruction scheme *generalized* ADMM. Our main contributions are summarized as follows:

- Proposal of a new estimator by learning a single convex penalty function whose proximal operator is applied to every iteration of the generalized ADMM. The convexity constraint is appropriately characterized in terms of the spline coefficients that parameterize the corresponding proximal operator. The learning process optimizes the coefficients so that the mean ℓ_2-normed error between a set of ground-truth signals and the ADMM reconstructions (from their noise-added versions) is minimized.
- Convergence proof of the generalized ADMM scheme based on the above-mentioned convexity confinement. Consequently, the learned penalty function is adjusted from one level of noise to another by a simple scaling operation, eliminating the need for retraining. Furthermore, assuming symmetrically distributed signals, the number of learning parameters is reduced by a half.
- Application of the proposed learning framework on two model signals, namely the Brownian motion and compound Poisson process. The main reason for choosing these models is that their (optimal) MMSE estimations are available for comparison. Furthermore, since these stochastic processes can be decorrelated by the finite difference operator, dictionary learning is no longer needed and we can focus only on the nonlinearity learning. Experiments show that, for a wide range of noise variances, ADMM reconstructions with learned penalty functions are almost identical to the MMSE estimators of these signals. We further demonstrate the practical advantages of the proposed learning scheme over its unconstrained counterpart.

C. Outline

In the sequel, we provide an overview of the necessary mathematical tools in Section II. In Section III, we present our spline-based parametrization for the proximal operator and formulate the unconstrained version of our algorithm. This is then followed by the introduction of the constraint formulation in terms of the spline coefficients in Section IV. We prove the convergence

and the scalability (with respect to noise power) in Section V. Finally, numerical results are illustrated in Section VI where we show that our algorithm achieves the MMSE performance for Lévy processes with different sparsity characteristics.

II. Background

A. Monotone Operator Theory

We review here some notation and background from convex analysis and monotone operator theory; see [38] for further details. Let us restrict ourselves to the Hilbert space $\mathcal{H} = \mathbb{R}^d$, for some dimension $d \geq 1$, equipped with the Euclidean scalar product $\langle \cdot\,, \cdot \rangle$ and norm $\| \cdot \|_2$. The identity operator on $\mathcal{H}$ is denoted by Id. Consider a set-valued operator $T : \mathcal{H} \to 2^{\mathcal{H}}$ that maps each vector $\boldsymbol{x} \in \mathcal{H}$ to a set $T\boldsymbol{x} \subset \mathcal{H}$. The domain, range, and graph of operator T are respectively defined by

$$\begin{aligned} \operatorname{dom} T &= \{\boldsymbol{x} \in \mathcal{H} \mid T\boldsymbol{x} \neq \emptyset\}, \\ \operatorname{ran} T &= \{\boldsymbol{u} \in \mathcal{H} \mid (\exists\, \boldsymbol{x} \in \mathcal{H})\, \boldsymbol{u} \in T\boldsymbol{x}\}, \\ \operatorname{gra} T &= \{(\boldsymbol{x}, \boldsymbol{u}) \in \mathcal{H} \times \mathcal{H} \mid \boldsymbol{u} \in T\boldsymbol{x}\}. \end{aligned}$$

We say that T is single-valued if $T\boldsymbol{x}$ has a unique element for all $\boldsymbol{x} \in \operatorname{dom} T$. The inverse T^{-1} of T is also a set-valued operator from $\mathcal{H}$ to $2^{\mathcal{H}}$ defined by

$$T^{-1}\boldsymbol{u} = \{\boldsymbol{x} \in \mathcal{H} \mid \boldsymbol{u} \in T\boldsymbol{x}\}.$$

It is straightforward that to see that $\operatorname{dom} T = \operatorname{ran} T^{-1}$ and $\operatorname{ran} T = \operatorname{dom} T^{-1}$. T is called *monotone* if

$$\langle \boldsymbol{x} - \boldsymbol{y}\,,\, \boldsymbol{u} - \boldsymbol{v} \rangle \geq 0, \quad \forall (\boldsymbol{x}, \boldsymbol{u}) \in \operatorname{gra} T, \forall (\boldsymbol{y}, \boldsymbol{v}) \in \operatorname{gra} T.$$

In the one-dimensional (1-D) case when $d = 1$, a monotone operator is simply a non-decreasing function. T is *maximally monotone* if it is monotone and there exists no monotone operator S such that $\operatorname{gra} T \subsetneq \operatorname{gra} S$. A handy characterization of the maximal monotonicity is given by Minty's theorem [38, Theorem 21.1].

Theorem 1 (Minty): A monotone operator $T : \mathcal{H} \to 2^{\mathcal{H}}$ is maximally monotone if and only if $\operatorname{ran}(\mathrm{Id} + T) = \mathcal{H}$.

For an integer $n \geq 2$, $T : \mathcal{H} \to 2^{\mathcal{H}}$ is *n-cyclically monotone* if, for every n points $(\boldsymbol{x}_i, \boldsymbol{u}_i) \in \operatorname{gra} T, i = 1, \ldots, n$, and for $x_{n+1} = x_1$, we have that

$$\sum_{i=1}^{n} \langle \boldsymbol{x}_{i+1} - \boldsymbol{x}_i\,,\, \boldsymbol{u}_i \rangle \leq 0.$$

An operator is cyclically monotone if it is n-cyclically monotone for all $n \geq 2$. This is a stronger notion of monotonicity because being monotone is equivalent to being 2-cyclically monotone. Moreover, T is *maximally cyclically monotone* if it is cyclically monotone and there exists no cyclically monotone operator S such that $\operatorname{gra} T \subsetneq \operatorname{gra} S$. An operator T is said to be *firmly nonexpansive* if

$$\langle \boldsymbol{x} - \boldsymbol{y}, \boldsymbol{u} - \boldsymbol{v} \rangle \geq \|\boldsymbol{u} - \boldsymbol{v}\|^2, \forall (\boldsymbol{x}, \boldsymbol{u}) \in \operatorname{gra} T, \forall (\boldsymbol{y}, \boldsymbol{v}) \in \operatorname{gra} T.$$

It is not difficult to see that a firmly nonexpansive operator must be both single-valued and monotone.

We denote by $\Gamma_0(\mathcal{H})$ the class of all proper lower-semicontinuous *convex* functions $f : \mathcal{H} \to (-\infty, +\infty]$. For any proper function $f : \mathcal{H} \to (-\infty, +\infty]$, the *subdifferential* operator $\partial f : \mathcal{H} \to 2^{\mathcal{H}}$ is defined by

$$\partial f(\boldsymbol{x}) = \{\boldsymbol{u} \in \mathcal{H} \mid \langle \boldsymbol{y} - \boldsymbol{x}, \boldsymbol{u} \rangle \leq f(\boldsymbol{y}) - f(\boldsymbol{x}), \forall \boldsymbol{y} \in \mathcal{H}\},$$

whereas, the *proximal* operator $\operatorname{prox}_f : \mathcal{H} \to 2^{\mathcal{H}}$ is given by

$$\operatorname{prox}_f(\boldsymbol{x}) = \operatorname*{argmin}_{\boldsymbol{u} \in \mathcal{H}} \left\{ f(\boldsymbol{u}) + \frac{1}{2}\|\boldsymbol{u} - \boldsymbol{x}\|_2^2 \right\}.$$

It is remarkable that, when $f \in \Gamma_0(\mathcal{H})$, ∂f is maximally cyclically monotone, prox_f is firmly nonexpansive, and the two operators are related by

$$\operatorname{prox}_f = (\mathrm{Id} + \partial f)^{-1}, \tag{3}$$

where the right-hand side is also referred to as the *resolvent* of ∂f. Interestingly, any maximally cyclically monotone operator is the subdifferential of some convex function, according to Rockafellar's theorem [38, Theorem 22.14].

Theorem 2 (Rockafellar): $A : \mathcal{H} \to 2^{\mathcal{H}}$ is maximally cyclically monotone if and only if there exists $f \in \Gamma_0(\mathcal{H})$ such that $A = \partial f$.

B. Denoising Problem and ADMM

Let us consider throughout this paper the denoising problem in which a signal $\boldsymbol{x} \in \mathbb{R}^N$ is estimated from its corrupted version $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{n}$, where $\boldsymbol{n}$ is assumed to be additive white Gaussian noise (AWGN) of variance σ^2. An estimator of $\boldsymbol{x}$ from $\boldsymbol{y}$ is denoted by $\hat{\boldsymbol{x}}(\boldsymbol{y})$. We treat $\boldsymbol{x}$ as a random vector generated from the joint probability density function (pdf) p_X. It is assumed that $\boldsymbol{x}$ is *whitenable* by a matrix $\boldsymbol{L} \in \mathbb{R}^{N \times N}$ such that the transformed vector $\boldsymbol{u} = \boldsymbol{L}\boldsymbol{x}$ has independent and identically distributed (i.i.d.) entries. The joint pdf p_U of the so-called *innovation* $\boldsymbol{u}$ is therefore separable, i.e.,

$$p_U(\boldsymbol{u}) = \prod_{i=1}^{N} p_U(u_i),$$

where, for convenience, p_U is reused to denote the 1-D pdf of each component of $\boldsymbol{u}$. We define $\Phi_U(\boldsymbol{u}) = -\log p_U(\boldsymbol{u})$ as the *penalty function* of $\boldsymbol{u}$. This function is then separable in the sense that

$$\Phi_U(\boldsymbol{u}) = \sum_{i=1}^{N} \Phi_U(u_i),$$

where Φ_U is again used to denote the 1-D penalty function of each entry u_i.

The MMSE estimator, which is optimal if the ultimate goal is to minimize the expected squared error between the estimate $\hat{\boldsymbol{x}}$ and the original signal $\boldsymbol{x}$, is given by Stein's formula [19]

$$\hat{\boldsymbol{x}}_{\mathrm{MMSE}}(\boldsymbol{y}) = \boldsymbol{y} + \sigma^2 \nabla \log p_Y(\boldsymbol{y}), \tag{4}$$

where p_Y is the joint pdf of the measurement $\boldsymbol{y}$ and ∇ denotes the gradient operator. Despite its elegant expression, the MMSE estimator, in most cases, is computationally intractable since p_Y is obtained through a high-dimensional convolution between the prior distribution p_X and the Gaussian distribution $g_\sigma(\boldsymbol{n}) = (2\pi\sigma^2)^{-N/2} \exp(-\|\boldsymbol{n}\|^2/2\sigma^2)$. However, for Lévy processes, which have independent and stationary increments, the MMSE estimator is computable using a message passing algorithm [39].

On the other hand, the MAP is given by

$$\begin{aligned}\hat{\boldsymbol{x}}_{\mathrm{MAP}}(\boldsymbol{y}) &= \operatorname*{argmax}_{\boldsymbol{x}} p_{X|Y}(\boldsymbol{x}|\boldsymbol{y}) \\ &= \operatorname*{argmax}_{\boldsymbol{x}} \left\{p_{Y|X}(\boldsymbol{y}|\boldsymbol{x})\, p_X(\boldsymbol{x})\right\} \\ &= \operatorname*{argmin}_{\boldsymbol{x}} \left\{\frac{1}{2}\|\boldsymbol{y}-\boldsymbol{x}\|_2^2 + \sigma^2 \Phi_X(\boldsymbol{x})\right\}. \quad (5)\end{aligned}$$

where $\Phi_X(\boldsymbol{x}) = -\log p_X(\boldsymbol{x})$ is the (nonseparable) penalty function of $\boldsymbol{x}$. In other words, the MAP estimator is exactly the proximal operator of $\sigma^2\Phi_X$. Assuming that the mapping $\boldsymbol{u} = \boldsymbol{L}\boldsymbol{x}$ is one-to-one, the minimization in (5) can be equivalently written as [24, p. 254]

$$\min_{\boldsymbol{x}} \left\{\frac{1}{2}\|\boldsymbol{y}-\boldsymbol{x}\|_2^2 + \sigma^2 \sum_{i=1}^{N} \Phi_U([\boldsymbol{L}\boldsymbol{x}]_i)\right\}. \quad (6)$$

This expression of the MAP reconstruction resembles the regularization-based approach (e.g., total variation method) in which the transform $\boldsymbol{L}$ is designed to sparsify the signal, the penalty function Φ_U is chosen—the typical choice being the ℓ_1-norm—to promote the sparsity of the transform coefficients $\boldsymbol{u}$. The parameter σ^2 is set (not necessarily to the noise variance) to trade off the quadratic fidelity term with the regularization term. The optimization problem (6) can be solved efficiently by iterative algorithms such as the alternating direction method of multipliers (ADMM) [16]. To that end, we form the augmented Langrangian

$$\frac{1}{2}\|\boldsymbol{y}-\boldsymbol{x}\|_2^2 + \sigma^2\Phi_U(\boldsymbol{u}) - \langle \boldsymbol{\alpha}, \boldsymbol{L}\boldsymbol{x}-\boldsymbol{u}\rangle + \frac{\mu}{2}\|\boldsymbol{L}\boldsymbol{x}-\boldsymbol{u}\|_2^2$$

and successively mimimize this functional with respect to each of the variables $\boldsymbol{x}$ and $\boldsymbol{u}$, while fixing the other one; the Lagrange multiplier $\boldsymbol{\alpha}$ is also updated appropriately at each step. In particular, at iteration $k+1$, the updates look like

$$\boldsymbol{x}^{(k+1)} = \left(\boldsymbol{I} + \mu \boldsymbol{L}^T\boldsymbol{L}\right)^{-1}\left(\boldsymbol{y} + \boldsymbol{L}^T\left(\mu\boldsymbol{u}^{(k)} + \boldsymbol{\alpha}^{(k)}\right)\right) \quad (7)$$

$$\boldsymbol{\alpha}^{(k+1)} = \boldsymbol{\alpha}^{(k)} - \mu\left(\boldsymbol{L}\boldsymbol{x}^{(k+1)} - \boldsymbol{u}^{(k)}\right) \quad (8)$$

$$\boldsymbol{u}^{(k+1)} = \mathrm{prox}_{\sigma^2/\mu\Phi_U}\left(\boldsymbol{L}\boldsymbol{x}^{(k+1)} - \frac{1}{\mu}\boldsymbol{\alpha}^{(k+1)}\right). \quad (9)$$

Here, $\boldsymbol{u}$ and $\boldsymbol{\alpha}$ are initialized to be $\boldsymbol{u}^{(0)}$ and $\boldsymbol{\alpha}^{(0)}$, respectively; $\boldsymbol{I} \in \mathbb{R}^{N\times N}$ denotes the identity matrix. If the proximal operator $\mathrm{prox}_{\sigma^2/\mu\Phi_U}$ in (9) is replaced with a general operator T, we refer to the above algorithm as the *generalized ADMM* associated with T. When the operator T is separable, i.e., $T(\boldsymbol{u}) = (T(u_1), \ldots, T(u_N))$, we refer to the 1-D function $T : \mathbb{R} \to \mathbb{R}$ as the *shrinkage function*; the name comes from the observation that typical proximal operators, such as the soft-thresholding, shrink large values of the input in a pointwise manner to reduce the noise. In what follows, we propose a learning approach to the denoising problem in which the shrinkage function T of the generalized ADMM is optimized in the MMSE sense from data, instead of being engineered as in sparsity-promoting schemes.

Algorithm 1: Unconstrained Learning.

Input: training example $(\boldsymbol{x}, \boldsymbol{y})$, learning rate $\gamma > 0$, sampling step Δ, number of spline knots $2M+1$.
Output: spline coefficients $\boldsymbol{c}^*$.

1: *Initialize*: Set $0 \leftarrow i$, choose $\boldsymbol{c}^{(0)} \in \mathbb{R}^{2M+1}$.
2: Compute the gradient $\nabla J\left(\boldsymbol{c}^{(i)}\right)$ via Algorithm 2.
3: Update $\boldsymbol{c}$ as:
$$\boldsymbol{c}^{(i+1)} = \boldsymbol{c}^{(i)} - \gamma \nabla J\left(\boldsymbol{c}^{(i)}\right).$$
4: Return $\boldsymbol{c}^* = \boldsymbol{c}^{(i+1)}$ if a stopping criterion is met, otherwise set $i \leftarrow i+1$ and go to step 2.

III. Learning Unconstrained Shrinkage Functions

A. Learning Algorithm

To learn the shrinkage function $T : \mathbb{R} \to \mathbb{R}$, we parameterize it via a spline representation:

$$T(x) = \sum_{m=-M}^{M} c_m \psi\left(\frac{x}{\Delta} - m\right), \quad (10)$$

where ψ is some kernel (radial basis functions, B-splines, etc.) and Δ is the sampling step size that defines the distant between consecutive spline knots. We call such function T a *Spline-Prox*. Consider a generalized ADMM using shrinkage function T associated with varying spline coefficients $\boldsymbol{c}$ while fixing the kernel ψ and other parameters of the algorithm (the transform $\boldsymbol{L}$, the penalty parameter μ, the initialization $\boldsymbol{x}^{(0)}$, and the number of iterations K). Therefore, the output $\boldsymbol{x}^{(K)}$ of the generalized ADMM is just a function of the spline coefficients $\boldsymbol{c}$ and the observation $\boldsymbol{y}$. Given a collection of ground-truth signals $\{\boldsymbol{x}_\ell\}_{\ell=1}^L$ and their observations $\{\boldsymbol{y}_\ell\}_{\ell=1}^L$, the vector $\boldsymbol{c} \in \mathbb{R}^{2M+1}$ is to be learned via minimizing the following cost function:

$$J(\boldsymbol{c}) = \frac{1}{2}\sum_{\ell=1}^{L}\left\|\boldsymbol{x}^{(K)}(\boldsymbol{c}, \boldsymbol{y}_\ell) - \boldsymbol{x}_\ell\right\|_2^2. \quad (11)$$

For notational simplicity, from now on we drop the subscript ℓ and develop a learning algorithm for a single training example $(\boldsymbol{x}, \boldsymbol{y})$ that can be easily generalized to training sets of arbitrary size. The cost function is thus simplified to

$$J(\boldsymbol{c}) = \frac{1}{2}\left\|\boldsymbol{x}^{(K)}(\boldsymbol{c}, \boldsymbol{y}) - \boldsymbol{x}\right\|_2^2. \quad (12)$$

Although nonconvex, this cost function is differentiable as long as the spline kernel ψ is differentiable. This allows us to carry out a simple gradient descent that is described in Algorithm 1. This algorithm serves as an intermediate step toward the constrained learning presented later in the paper and, thus, is named unconstrained learning. As in every neural network, the computation of the gradient of the cost function is performed in a backpropagation manner, which will be detailed in the next section.

B. Gradient Computation

We devise in this section a backpropagation algorithm to evaluate the gradient of the cost function with respect to the spline

coefficients of the shrinkage function. We adopt the following convention for matrix calculus: for a function $y : \mathbb{R}^m \to \mathbb{R}$ of vector variable $\boldsymbol{x}$, its gradient is a column vector given by

$$\nabla y(\boldsymbol{x}) = \frac{\partial y}{\partial \boldsymbol{x}} = \left[\frac{\partial y}{\partial x_1} \quad \frac{\partial y}{\partial x_2} \quad \cdots \quad \frac{\partial y}{\partial x_m}\right]^T,$$

whereas, for a vector-valued function $\boldsymbol{y} : \mathbb{R}^m \to \mathbb{R}^n$ of vector variable $\boldsymbol{x}$, its Jacobian is an $m \times n$ matrix defined by

$$\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}} = \left[\frac{\partial y_1}{\partial \boldsymbol{x}} \quad \frac{\partial y_2}{\partial \boldsymbol{x}} \quad \cdots \quad \frac{\partial y_n}{\partial \boldsymbol{x}}\right] = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_n}{\partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_1}{\partial x_m} & \cdots & \frac{\partial y_n}{\partial x_m} \end{bmatrix}.$$

We are now ready to compute the gradient of the cost function J with respect to the parameter vector $\boldsymbol{c}$. For simplicity, for $k = 0, \ldots, K-1$, put

$$\begin{aligned} \boldsymbol{M} &= (\boldsymbol{I} + \mu \boldsymbol{L}^T \boldsymbol{L})^{-1}, \\ \boldsymbol{z}^{(k+1)} &= \boldsymbol{y} + \boldsymbol{L}^T \left(\mu \boldsymbol{u}^{(k)} + \boldsymbol{\alpha}^{(k)}\right), \\ \boldsymbol{v}^{(k+1)} &= \boldsymbol{L}\boldsymbol{x}^{(k+1)} - \frac{1}{\mu}\boldsymbol{\alpha}^{(k+1)}. \end{aligned}$$

By using these notations, we concisely write the updates at iteration $k+1$ of the generalized ADMM associated with operator T as

$$\begin{aligned} \boldsymbol{x}^{(k+1)} &= \boldsymbol{M}\boldsymbol{z}^{(k+1)}, \\ \boldsymbol{\alpha}^{(k+1)} &= \boldsymbol{\alpha}^{(k)} - \mu\left(\boldsymbol{L}\boldsymbol{x}^{(k+1)} - \boldsymbol{u}^{(k)}\right), \\ \boldsymbol{u}^{(k+1)} &= T\left(\boldsymbol{v}^{(k+1)}\right). \end{aligned}$$

First, applying the chain rule to (12) yields

$$\nabla J(\boldsymbol{c}) = \frac{\partial \boldsymbol{x}^{(K)}}{\partial \boldsymbol{c}} \frac{\partial J}{\partial \boldsymbol{x}^{(K)}} = \frac{\partial \boldsymbol{x}^{(K)}}{\partial \boldsymbol{c}} \left(\boldsymbol{x}^{(K)} - \boldsymbol{x}\right) \tag{13}$$

Next, from the updates of the ADMM and by noting that $\boldsymbol{L}$ and $\boldsymbol{y}$ does not depend on $\boldsymbol{c}$, for $k = 0, \ldots, K-1$, we get

$$\begin{aligned} \frac{\partial \boldsymbol{x}^{(k+1)}}{\partial \boldsymbol{c}} &= \frac{\partial \boldsymbol{z}^{(k+1)}}{\partial \boldsymbol{c}} \boldsymbol{M}^T = \left(\mu \frac{\partial \boldsymbol{u}^{(k)}}{\partial \boldsymbol{c}} + \frac{\partial \boldsymbol{\alpha}^{(k)}}{\partial \boldsymbol{c}}\right) \boldsymbol{L}\boldsymbol{M}, \\ \frac{\partial \boldsymbol{\alpha}^{(k)}}{\partial \boldsymbol{c}} &= \frac{\partial \boldsymbol{\alpha}^{(k-1)}}{\partial \boldsymbol{c}} - \mu \frac{\partial \boldsymbol{x}^{(k)}}{\partial \boldsymbol{c}} \boldsymbol{L}^T + \mu \frac{\partial \boldsymbol{u}^{(k-1)}}{\partial \boldsymbol{c}}, \end{aligned}$$

and

$$\begin{aligned} \frac{\partial \boldsymbol{u}^{(k)}}{\partial \boldsymbol{c}} &= \frac{\partial \boldsymbol{v}^{(k)}}{\partial \boldsymbol{c}} \frac{\partial \boldsymbol{u}^{(k)}}{\partial \boldsymbol{v}^{(k)}} + \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{c}} \frac{\partial \boldsymbol{u}^{(k)}}{\partial \boldsymbol{c}} \\ &= \left(\frac{\partial \boldsymbol{x}^{(k)}}{\partial \boldsymbol{c}} \boldsymbol{L}^T - \frac{1}{\mu} \frac{\partial \boldsymbol{\alpha}^{(k)}}{\partial \boldsymbol{c}}\right) \boldsymbol{D}^{(k)} + \boldsymbol{\Psi}^{(k)}, \end{aligned}$$

where $\boldsymbol{D}^{(k)} = \operatorname{diag}\left(T'(\boldsymbol{v}^{(k)})\right)$ is the diagonal matrix whose entries on the diagonal are the derivatives of T at $\{v_i^{(k)}\}_{i=1}^N$, and $\boldsymbol{\Psi}^{(k)}$ is a matrix defined by

$$\Psi_{ij}^{(k)} = \psi\left(\frac{v_j^{(k)}}{\Delta} - i\right).$$

Algorithm 2: Backpropagation for Unconstrained Learning.

Input: signal $\boldsymbol{x} \in \mathbb{R}^N$, measurement $\boldsymbol{y} \in \mathbb{R}^N$, transform matrix $\boldsymbol{L} \in \mathbb{R}^{N \times N}$, kernel ψ, sampling step Δ, number of spline knots $2M+1$, current spline coefficients $\boldsymbol{c} \in \mathbb{R}^{2M+1}$, number of ADMM iterations K.

Output: gradient $\nabla J(\boldsymbol{c})$.

1: Define:

$$\psi_i = \psi(\cdot/\Delta - i), \text{ for } i = -M, \ldots, M$$

$$\boldsymbol{A} = \boldsymbol{L}\left(\boldsymbol{I} + \mu \boldsymbol{L}^T \boldsymbol{L}\right)^{-1}$$

2: Run K iterations of the generalized ADMM with the SplineProx $T = \sum_{i=-M}^{M} c_i \psi_i$. Store $\boldsymbol{x}^{(K)}$ and, for all $k = 1, \ldots, K$, store

$$\boldsymbol{v}^{(k)} = \boldsymbol{L}\boldsymbol{x}^{(k)} - \boldsymbol{\alpha}^{(k)}/\mu,$$

$$\boldsymbol{\Psi}^{(k)} = \left\{\psi_i\left(v_j^{(k)}\right)\right\}_{i,j},$$

$$\boldsymbol{B}^{(k)} = \boldsymbol{I} - \mu \boldsymbol{A}\boldsymbol{L}^T + \left(2\mu \boldsymbol{A}\boldsymbol{L}^T - \boldsymbol{I}\right) \operatorname{diag}(T'(\boldsymbol{v}^{(k)})).$$

3: *Initialize*: $\boldsymbol{r} = \boldsymbol{A}(\boldsymbol{x}^{(K)} - \boldsymbol{x}), \boldsymbol{g} = \boldsymbol{0}, k = K-1$.

4: Compute:

$$\boldsymbol{g} \leftarrow \boldsymbol{g} + \mu \boldsymbol{\Psi}^{(k)} \boldsymbol{r},$$

$$\boldsymbol{r} \leftarrow \boldsymbol{B}^{(k)} \boldsymbol{r}.$$

5: If $k = 1$, return $\nabla J(\boldsymbol{c}) = \boldsymbol{g}$, otherwise, set $k \leftarrow k-1$ and go to step 4.

Proceeding with simple algebraic manipulation, we arrive at

$$\frac{\partial \boldsymbol{x}^{(k+1)}}{\partial \boldsymbol{c}} = \left(\frac{\partial \boldsymbol{\alpha}^{(k)}}{\partial \boldsymbol{c}} + \mu \frac{\partial \boldsymbol{u}^{(k)}}{\partial \boldsymbol{c}}\right) \boldsymbol{A}, \tag{14a}$$

$$\frac{\partial \boldsymbol{\alpha}^{(k)}}{\partial \boldsymbol{c}} + \mu \frac{\partial \boldsymbol{u}^{(k)}}{\partial \boldsymbol{c}} = \left(\frac{\partial \boldsymbol{\alpha}^{(k-1)}}{\partial \boldsymbol{c}} + \mu \frac{\partial \boldsymbol{u}^{(k-1)}}{\partial \boldsymbol{c}}\right) \boldsymbol{B}^{(k)} + \mu \boldsymbol{\Psi}^{(k)}. \tag{14b}$$

where

$$\begin{aligned} \boldsymbol{A} &= \boldsymbol{L}\left(\boldsymbol{I} + \mu \boldsymbol{L}^T \boldsymbol{L}\right)^{-1}, \\ \boldsymbol{B}^{(k)} &= \boldsymbol{I} - \mu \boldsymbol{A}\boldsymbol{L}^T + \left(2\mu \boldsymbol{A}\boldsymbol{L}^T - \boldsymbol{I}\right) \boldsymbol{D}^{(k)}. \end{aligned}$$

Finally, by combining (13) with (14) and by noting that $\partial \boldsymbol{\alpha}^{(0)}/\partial \boldsymbol{c} = \partial \boldsymbol{u}^{(0)}/\partial \boldsymbol{c} = \boldsymbol{0}$, we propose a backpropagation algorithm to compute the gradient of the cost function J with respect to the spline coefficients $\boldsymbol{c}$ as described in Algorithm 2. We refer to the generalized ADMM that uses a shrinkage function learned via Algorithm 1 as MMSE-ADMM.

IV. Learning Constrained Shrinkage Functions

We propose two constraints for learning the shrinkage functions: firm nonexpansiveness and antisymmetry. The former is motivated by the well-known fact that the proximal operator of a convex function must be firmly nonexpansive [38]; the latter is justified by Theorem 3: symmetrically distributed signals imply antisymmetric proximal operator and vice versa.

Theorem 3: Let $\Phi \in \Gamma_0(\mathbb{R}^N)$. Φ is symmetric if and only if prox_Φ is antisymmetric.

Proof: See the appendix. ■

In order to incorporate the firmly nonexpansive constraint into the learning of spline coefficients, we choose the kernel ψ in the representation (10) to be a B-spline of some integer order. Recall that the B-spline β^n of integer order $n \geq 0$ is defined recursively as

$$\beta^0(x) = \begin{cases} 1, & |x| \leq 1/2 \\ 0, & |x| > 1/2, \end{cases}$$
$$\beta^n = \beta^{n-1} * \beta^0, \quad n \geq 1.$$

We adopt these type of kernels because they are compactly supported and their derivatives can be simply expressed in terms of B-splines of smaller orders [40]. These computational advantages help speedup the learning process. More importantly, as pointed out in Theorem 4, by using B-spline kernels, the firm nonexpansiveness of a SplineProx is satisfied as long as its coefficients obey a simple linear constraint.

Theorem 4: Let $\Delta > 0$ and let β^n be the B-spline of order $n \geq 1$. If c is a sequence such that $0 \leq c_m - c_{m-1} \leq \Delta, \forall m \in \mathbb{Z}$, then $f = \sum_{m\in\mathbb{Z}} c_m \beta^n(\cdot/\Delta - m)$ is a firmly nonexpansive function.

Proof: Since f is a 1-D function, it is easy to see that f is firmly nonexpansive if and only if

$$0 \leq f(x) - f(y) \leq x - y, \quad \forall x > y. \tag{15}$$

We now show (15) by considering 2 different cases.

$\underline{n = 1:}$ β^1 is the triangle function, and so f is continuous and piecewise-linear. If $x, y \in [(m-1)\Delta, m\Delta]$ for some $m \in \mathbb{Z}$, then

$$0 \leq \frac{f(x) - f(y)}{x - y} = \frac{c_m - c_{m-1}}{\Delta} \leq 1,$$

which implies

$$0 \leq f(x) - f(y) \leq x - y, \quad \forall (m-1)\Delta \leq y < x \leq m\Delta. \tag{16}$$

Otherwise, there exist $k, \ell \in \mathbb{Z}$ such that $y \in [(k-1)\Delta, k\Delta]$, $x \in [\ell\Delta, (\ell+1)\Delta]$. Then, we write

$$f(x) - f(y) = [f(x) - f(\ell\Delta)] + [f(\ell\Delta) - f((\ell-1)\Delta)] + \cdots + [f((\ell+1)\Delta) - f(\ell\Delta)] + [f(k) - f(y)].$$

By applying (16) to each term of the above sum, we obtain the desired pair of inequalities in (15), which implies the firm nonexpansiveness of f.

$\underline{n \geq 2:}$ β^n is now differentiable and so is f. Thus, by using the mean value theorem, (15) is achieved if the derivative f' of f is bounded between 0 and 1, which will be shown subsequently. Recall that the derivative of β^n is equal to the finite difference of β^{n-1}. In particular,

$$(\beta^n)'(x) = \beta^{n-1}\left(x + \frac{1}{2}\right) - \beta^{n-1}\left(x - \frac{1}{2}\right). \tag{17}$$

Algorithm 3: Constrained Learning.

Input: training example $(\boldsymbol{x}, \boldsymbol{y})$, learning rate $\gamma > 0$, sampling step Δ, number of spline knots $2M + 1$.
Output: spline coefficients $\boldsymbol{c}^*$.

1: Define the linear constraint set
$$\mathcal{S} = \left\{\boldsymbol{c} \in \mathbb{R}^M \,|\, 0 \leq c_m - c_{m-1} \leq \Delta, \forall m = 2, \ldots, M\right\}$$
2: *Initialize*: Set $0 \leftarrow i$, choose $\boldsymbol{c}^{(0)} \in \mathcal{S}$.
3: Compute the gradient $\nabla J\left(\boldsymbol{c}^{(i)}\right)$ via Algorithm 2.
4: Update $\boldsymbol{c}$ as:
$$\boldsymbol{c}^{(i+1)} = \mathrm{proj}_{\mathcal{S}}\left(\boldsymbol{c}^{(i)} - \gamma \nabla J\left(\boldsymbol{c}^{(i)}\right)\right).$$
5: Return $\boldsymbol{c}^* = \boldsymbol{c}^{(i+1)}$ if a stopping criterion is met, otherwise set $i \leftarrow i + 1$ and go to step 3.

Hence, for all $x \in \mathbb{R}$,

$$\begin{aligned} f'(x) &= \frac{1}{\Delta}\sum_{m\in\mathbb{Z}} c_m (\beta^n)'\left(\frac{x}{\Delta} - m\right) \\ &= \frac{1}{\Delta}\sum_{m\in\mathbb{Z}} c_m \left\{\beta^{n-1}\left(\frac{x}{\Delta} - m + \frac{1}{2}\right) - \beta^{n-1}\left(\frac{x}{\Delta} - m - \frac{1}{2}\right)\right\} \\ &= \frac{1}{\Delta}\sum_{m\in\mathbb{Z}} c_m \beta^{n-1}\left(\frac{x}{\Delta} - m + \frac{1}{2}\right) \\ &\quad - \frac{1}{\Delta}\sum_{m\in\mathbb{Z}} c_{m-1} \beta^{n-1}\left(\frac{x}{\Delta} - m + \frac{1}{2}\right) \text{ (change of variable)} \\ &= \frac{1}{\Delta}\sum_{m\in\mathbb{Z}} (c_m - c_{m-1}) \beta^{n-1}\left(\frac{x}{\Delta} + \frac{1}{2} - m\right). \end{aligned}$$

Since $0 \leq c_m - c_{m-1} \leq \Delta, \forall m \in \mathbb{Z}$ and since $\beta^{n-1}(x/\Delta + 1/2 - m) \geq 0, \forall x \in \mathbb{R}, m \in \mathbb{Z}$, one has the following pair of inequalities for all $x \in \mathbb{R}$:

$$0 \leq f'(x) \leq \sum_{m\in\mathbb{Z}} \beta^{n-1}\left(\frac{x}{\Delta} + \frac{1}{2} - m\right). \tag{18}$$

By using the partition-of-unity property of the B-spline β^{n-1}, (18) is simplified to

$$0 \leq f'(x) \leq 1, \quad \forall x \in \mathbb{R},$$

which finally proves that f is a firmly nonexpansive function. ■

With the above results, we easily design an algorithm for learning antisymmetric and firmly nonexpansive shrinkage functions. Algorithm 3, the main focus of this paper, is the constrained counterpart of Algorithm 1: the gradient descent is replaced with a projected gradient descent where, at each update, the spline coefficients are projected onto the linear-constraint set described in Theorem 4 (this projection is performed via a quadratic programming). The gradient of the cost function in this case is evaluated through Algorithm 4, which is just slightly modified from Algorithm 2 to adapt to the antisymmetric nature of the SplineProx. Specifically, we assume that the spline

Algorithm 4: Backpropagation for Constrained Learning.

Input: signal $\boldsymbol{x} \in \mathbb{R}^N$, measurement $\boldsymbol{y} \in \mathbb{R}^N$, transform matrix $\boldsymbol{L} \in \mathbb{R}^{N \times N}$, B-spline $\psi = \beta^n$, sampling step Δ, number of spline knots $2M+1$, current spline coefficients $\boldsymbol{c} \in \mathbb{R}^M$, number of ADMM iterations K.
Output: gradient $\nabla J(\boldsymbol{c})$.

1: Define:
$$\tilde{\psi}_i = \psi(\cdot/\Delta - i) - \psi(\cdot/\Delta + i), \text{ for } i = 1, \ldots, M$$
$$\boldsymbol{A} = \boldsymbol{L}\left(\boldsymbol{I} + \mu \boldsymbol{L}^T \boldsymbol{L}\right)^{-1}.$$
2: Run K iterations of the generalized ADMM with the *antisymmetric* SplineProx $T = \sum_{i=1}^{M} c_i \tilde{\psi}_i$. Store $\boldsymbol{x}^{(K)}$ and, for all $k = 1, \ldots, K$, store
$$\boldsymbol{v}^{(k)} = \boldsymbol{L}\boldsymbol{x}^{(k)} - \boldsymbol{\alpha}^{(k)}/\mu,$$
$$\boldsymbol{\Psi}^{(k)} = \left\{\tilde{\psi}_i\left(v_j^{(k)}\right)\right\}_{i,j},$$
$$\boldsymbol{B}^{(k)} = \boldsymbol{I} - \mu \boldsymbol{A}\boldsymbol{L}^T + \left(2\mu \boldsymbol{A}\boldsymbol{L}^T - \boldsymbol{I}\right)\operatorname{diag}(T'(\boldsymbol{v}^{(k)})).$$
3: *Initialize*: $\boldsymbol{r} = \boldsymbol{A}^T(\boldsymbol{x}^{(K)} - \boldsymbol{x}), \boldsymbol{g} = \boldsymbol{0}, k = K-1$.
4: Compute:
$$\boldsymbol{g} \leftarrow \boldsymbol{g} + \mu \boldsymbol{\Psi}^{(k)} \boldsymbol{r},$$
$$\boldsymbol{r} \leftarrow \boldsymbol{B}^{(k)} \boldsymbol{r}.$$
5: If $k = 1$, return $\nabla J(\boldsymbol{c}) = \boldsymbol{g}$, otherwise, set $k \leftarrow k-1$ and go to step 4.

coefficients obey the relation $c_{-m} = -c_m$ and rewrite (10) as

$$T(x) = \sum_{m=1}^{M} c_m \left[\psi\left(\frac{x}{\Delta} - m\right) - \psi\left(\frac{x}{\Delta} + m\right)\right] = \sum_{m=1}^{M} c_m \tilde{\psi}_m(x), \tag{19}$$

where $\tilde{\psi}_m = \psi(\cdot/\Delta - m) - \psi(\cdot/\Delta + m)$ is an antisymmetric function. The rest of the gradient computation is similar to that of Algorithm 2. We refer to the generalized ADMM that uses a shrinkage function learned by Algorithm 3 as MMSE-CADMM, where the letter 'C' stands for 'convex.' The convexity of this learning scheme will be made clear in Section V.

V. Advantages of Adding Constraints

It is clear that imposing the antisymmetric constraint on the shrinkage function reduces the dimension of the optimization problem by a half and therefore substantially reduces the learning time. In this section, we demonstrate, from the theoretical point of view, the two important advantages of imposing the firmly nonexpansive constraint on the shrinkage function: convergence guarantee and scalability with noise level. Thanks to these properties, our learning-based denoiser behaves like a MAP estimator with some convex penalty function that is now different from the conventional penalty function. On the other hand, as experiments later show, the constrained learning scheme nearly achieves the optimal denoising performance of the MMSE estimator.

A. Convergence Guarantee

The following result asserts that the ADMM denoising converges, no matter what the noisy signal is, if it uses a separable firmly nonexpansive operator in the place of the conventional proximal operator. Interestingly, as will be shown in the proof, any separable firmly nonexpansive operator is the proximal operator of a separable convex penalty function. We want to emphasize that the separability is needed to establish this connection, although the reverse statement is known to hold in the multidimensional case [38].

Theorem 5: Let $\mu > 0$. If $T : \mathbb{R} \to \mathbb{R}$ is a 1-D firmly nonexpansive function such that $\operatorname{dom} T = \mathbb{R}$, then, for every input $\boldsymbol{y} \in \mathbb{R}^N$, the reconstruction sequence $\{\boldsymbol{x}^{(k)}\}$ of the generalized ADMM associated with the separable operator T and the penalty parameter μ converges to

$$\boldsymbol{x}^* = \underset{\boldsymbol{x} \in \mathbb{R}^N}{\operatorname{argmin}} \left\{\frac{1}{2}\|\boldsymbol{y} - \boldsymbol{x}\|_2^2 + \sum_{i=1}^{N} \Phi([\boldsymbol{L}\boldsymbol{x}]_i)\right\},$$

as $k \to \infty$, where $\Phi \in \Gamma_0(\mathbb{R})$ is a 1-D convex function such that $T = \operatorname{prox}_{\Phi/\mu}$.

Proof: First, we show that there exists a function $\Phi \in \Gamma_0(\mathbb{R})$ such that $T = \operatorname{prox}_{\Phi/\mu}$. To that end, let us define

$$S = (T^{-1} - \mathrm{Id}).$$

The firm nonexpansiveness of T then implies

$$\begin{aligned} &(x-y)(u-v) \geq (x-y)^2, && \forall u \in T^{-1}x, v \in T^{-1}y \\ \Leftrightarrow\ &(x-y)((u-x)-(v-y)) \geq 0, && \forall u \in T^{-1}x, v \in T^{-1}y \\ \Leftrightarrow\ &(x-y)(\tilde{u}-\tilde{v}) \geq 0, && \forall \tilde{u} \in Sx, \tilde{v} \in Sy, \end{aligned}$$

which means that S is a monotone operator. Furthermore, we have that

$$\operatorname{ran}(S + \mathrm{Id}) = \operatorname{ran}\left(T^{-1}\right) = \operatorname{dom} T = \mathbb{R}.$$

Therefore, S is maximally monotone thanks to Minty's theorem (Theorem 1). Since S is an operator on $\mathbb{R}$, we invoke [38, Thm. 22.18] to deduce that S must also be maximally cyclically monotone. Now, as a consequence of Rockafellar's theorem (Theorem 2), there exists a function $\tilde{\Phi} \in \Gamma_0(\mathbb{R})$ such that $\partial\tilde{\Phi} = S$. Let $\Phi = \mu\tilde{\Phi}$. Then, $\Phi \in \Gamma_0(\mathbb{R})$ and

$$T = (\mathrm{Id} + S)^{-1} = (\mathrm{Id} + \partial(\Phi/\mu))^{-1} = \operatorname{prox}_{\Phi/\mu}.$$

Next, define the cost function

$$f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{x}\|_2^2 + \sum_{i=1}^{N} \Phi([\boldsymbol{L}\boldsymbol{x}]_i). \tag{20}$$

Replacing T with $\operatorname{prox}_{\Phi/\mu}$, the generalized ADMM associated with T becomes the regular ADMM associated with the above cost function f. By using the convexity of the ℓ_2-norm and of the function Φ, it is well known [16, Section 3.2.1] that $f\left(\boldsymbol{x}^{(k)}\right) \to p^*$ as $k \to \infty$, where p^* is the minimum value of f.

Finally, notice that the function f defined in (20) is strongly convex. Thus, there exists a unique minimizer $\boldsymbol{x}^* \in \mathbb{R}^N$ such that $f(\boldsymbol{x}^*) = p^*$. Moreover, $\boldsymbol{x}^*$ is known to be a strong minimizer [42, Lemma 2.26] in the sense that the convergence of $\left\{f\left(\boldsymbol{x}^{(k)}\right)\right\}$ to $f(\boldsymbol{x}^*)$ implies the convergence of $\left\{\boldsymbol{x}^{(k)}\right\}$ to $\boldsymbol{x}^*$. This completes the proof. ■

B. Scalability With Noise Level

In all existing learning schemes, the shrinkage function is learned for a particular level of noise and then applied in the reconstruction of testing signals corrupted with the same level of noise. If the noise variance changes, the shrinkage function must be relearned from scratch, which will create a computational burden on top of the ADMM reconstruction. This drawback is due to the unconstrained learning strategy in which the shrinkage function is not necessarily the proximal operator of any function. Despite its flexibility, arbitrary nonlinearity no longer goes hand-in-hand with a regularization-based minimization (MAP-like estimation). By contrast, our constrained learning scheme maintains a connection with the underlying minimization regularized by a convex penalty function. This strategy allows us to easily adjust the learned shrinkage function from one level of noise to another by simply scaling the corresponding penalty function by the ratio between noise variances like the conventional MAP estimator. Proposition 1 provides a useful formula to extrapolate the proximal operator of a scaled convex function from the proximal operator of the original function.

Proposition 1: For all $f \in \Gamma_0(\mathbb{R}^N)$,

$$\operatorname{prox}_{\lambda f} = \left(\lambda \operatorname{prox}_f^{-1} + (1-\lambda)\operatorname{Id}\right)^{-1}, \quad \forall \lambda \geq 0. \tag{21}$$

Proof: Recall a basic result in convex analysis [42] that $\partial(\lambda f) = \lambda \partial f$, for all $f \in \Gamma_0(\mathbb{R}^N)$ and for all $\lambda \geq 0$. Also recall that the proximal operator of a convex function is the resolvent of the subdifferential operator. Therefore,

$$\begin{aligned}\operatorname{prox}_{\lambda f} &= (\operatorname{Id} + \partial(\lambda f))^{-1} = (\operatorname{Id} + \lambda \partial f)^{-1} \\ &= \left(\operatorname{Id} + \lambda\left(\operatorname{prox}_f^{-1} - \operatorname{Id}\right)\right)^{-1} \\ &= \left(\lambda \operatorname{prox}_f^{-1} + (1-\lambda)\operatorname{Id}\right)^{-1},\end{aligned}$$

completing the proof. ■

The next result establishes that all members of the family generated by (21) are firmly nonexpansive when the generator prox_f is replaced with a general firmly nonexpansive operator. It is noteworthy that the result holds in the multidimensional case where a firmly nonexpansive operator is not necessarily the proximal operator of a convex function.

Theorem 6: If $T : \mathbb{R}^N \to \mathbb{R}^N$ is a firmly nonexpansive operator such that $\operatorname{dom} T = \mathbb{R}^N$, then, for all $\lambda \geq 0$, $T_\lambda = (\lambda T^{-1} + (1-\lambda)\operatorname{Id})^{-1}$ is firmly nonexpansive and $\operatorname{dom} T_\lambda = \mathbb{R}^N$ as well.

Proof: The claim is trivial for $\lambda = 0$. Assume from now on that $\lambda > 0$. We first show that $\operatorname{dom} T_\lambda = \mathbb{R}^N$. By a straightforward extension of the argument in the proof of Theorem 5 to the multidimensional case, we easily have that the operator $S = T^{-1} - \operatorname{Id}$ is maximally monotone. It follows that λS is also maximally monotone for all $\lambda > 0$. By applying Minty's theorem to the operator λS, we obtain

$$\operatorname{dom} T_\lambda = \operatorname{ran}(\lambda T^{-1} + (1-\lambda)\operatorname{Id}) = \operatorname{ran}(\operatorname{Id} + \lambda S) = \mathbb{R}^N.$$

Next, we show that T_λ is firmly nonexpansive. Let $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^N$ and $\boldsymbol{u} \in T_\lambda(\boldsymbol{x}), \boldsymbol{v} \in T_\lambda(\boldsymbol{y})$. By the definition of T_λ, one readily verifies that

$$\boldsymbol{u} = T\left(\frac{\boldsymbol{x} + (\lambda-1)\boldsymbol{u}}{\lambda}\right), \quad \boldsymbol{v} = T\left(\frac{\boldsymbol{y} + (\lambda-1)\boldsymbol{v}}{\lambda}\right).$$

The firm nonexpansiveness of T yields

$$\begin{aligned}\|\boldsymbol{u} - \boldsymbol{v}\|^2 &\leq \left\langle \frac{\boldsymbol{x} + (\lambda-1)\boldsymbol{u}}{\lambda} - \frac{\boldsymbol{y} + (\lambda-1)\boldsymbol{u}}{\lambda}, \boldsymbol{u} - \boldsymbol{v} \right\rangle \\ &= \frac{\lambda-1}{\lambda}\|\boldsymbol{u} - \boldsymbol{v}\|^2 + \frac{1}{\lambda}\langle \boldsymbol{x} - \boldsymbol{y}, \boldsymbol{u} - \boldsymbol{v}\rangle,\end{aligned}$$

which translates to

$$\|\boldsymbol{u} - \boldsymbol{v}\|^2 \leq \langle \boldsymbol{x} - \boldsymbol{y}, \boldsymbol{u} - \boldsymbol{v}\rangle.$$

This confirms that T_λ is a firmly nonexpansive operator. ■

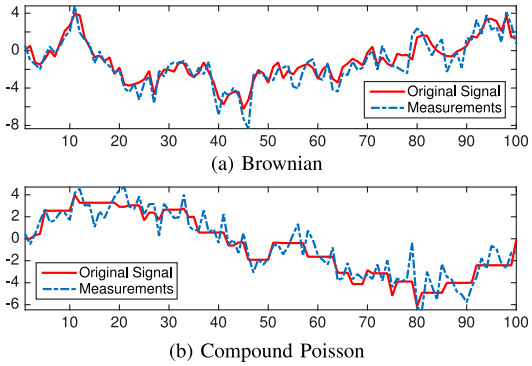


Fig. 1. Realizations of a Brownian motion and a compound Poisson process are plotted along with their corrupted versions with AWGN of variance $\sigma^2 = 1$.

VI. Experimental Results

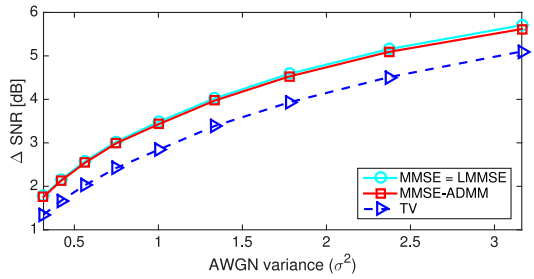
In this section, we report the experimental denoising results of the two proposed learning schemes: ADMM with *unconstrained* shrinkage functions learned via Algorithm 1 (denoted MMSE-ADMM) and ADMM with *constrained* shrinkage functions learned via Algorithm 3 (denoted MMSE-CADMM). Throughout this section, the transform $\boldsymbol{L}$ is fixed to be the finite difference operator: $[\boldsymbol{L}\boldsymbol{x}]_i = x_i - x_{(i-1) \bmod N}, \forall i$; the signal length is fixed to $N = 100$. Experiments were implemented in MATLAB on the two following types of Lévy processes:

1) Brownian motion: entries of the increment vector $\boldsymbol{u} = \boldsymbol{L}\boldsymbol{x}$ are i.i.d. Gaussian with unit variance: $p_U(u) = e^{-u^2/2}/\sqrt{2\pi}$.
2) Compound Poisson: entries of the increment vector $\boldsymbol{u} = \boldsymbol{L}\boldsymbol{x}$ are i.i.d. Bernoulli-Gaussian: $p_U(u) = (1 - e^{-\lambda})\, e^{-u^2/2}/\sqrt{2\pi} + e^{-\lambda}\delta(u)$, where δ is the Dirac impulse and $\lambda = 0.6$ is fixed. This results in a piecewise-constant signal $\boldsymbol{x}$ with Gaussian jumps.

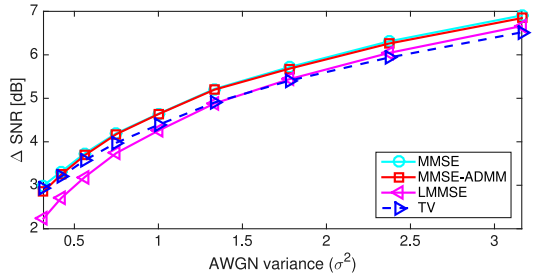
Specific realizations of these processes and their corrupted versions with AWGN of variance $\sigma^2 = 1$ are shown in Fig. 1.

A. Denoising Performance

The same parameters were chosen for both learning schemes (constrained and unconstrained). In particular, for each type of processes, a set of 500 signals was used for training and another set of 500 for testing. The number of ADMM layers (iterations) was set to $K = 10$; the penalty parameter of the

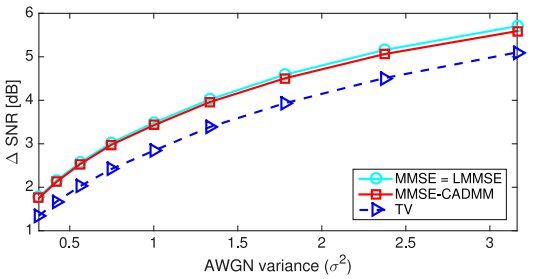


(a) Brownian

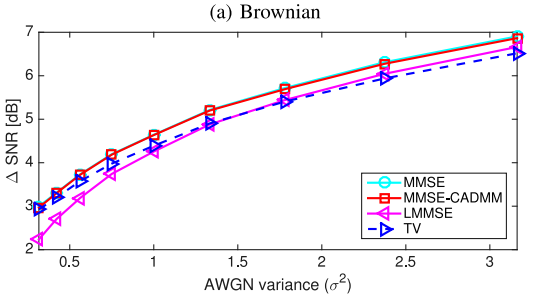


(b) Compound Poisson

Fig. 2. Denoising performances of the MMSE-ADMM where the unconstrained shrinkage functions are learned for all instances of the noise variance.



(a) Brownian



(b) Compound Poisson

Fig. 3. Denoising performances of the MMSE-CADMM where the constrained shrinkage functions are learned for all instances of the noise variance.

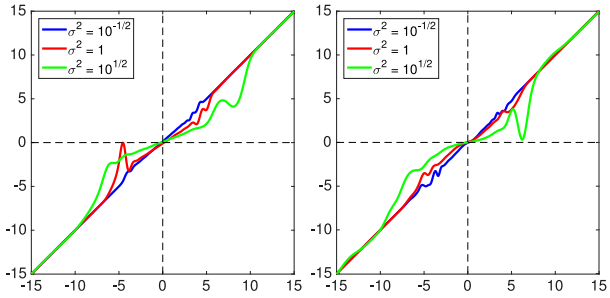
augmented Lagrangian was set to $\mu = 2$. The shrinkage function was represented with the cubic B-spline:

$$\psi(x) = \beta^3(x) = \begin{cases} \frac{2}{3} - |x|^2 + \frac{|x|^3}{2}, & 0 \le |x| < 1 \\ \frac{1}{6}(2 - |x|)^3, & 1 \le |x| < 2 \\ 0, & 2 \le |x|. \end{cases}$$

Cubic splines have the ability to approximate arbitrary functions and they are also known to offer the best cost/quality trade-off [43]. The spline coefficients $\{c_m\}$ were located uniformly in the dynamic range of $\boldsymbol{u} = \boldsymbol{L}\boldsymbol{x}$ with sampling step $\Delta = \sigma/2$, which is dependent on the noise level. Learning was performed by running either Algorithm 1 or Algorithm 3 for 1000 iterations with learning rate $\gamma = 2 \times 10^{-4}$. The shrinkage function was always initialized with the *identity* line, which corresponds to $c_m^{(0)} = m\Delta$ for all m.

The denoising performances were numerically evaluated by the signal-to-noise ratio (SNR) improvement that is defined by $\Delta\text{SNR}\,[\text{dB}] = 10\log_{10}\left(\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_2^2 / \|\boldsymbol{y} - \boldsymbol{x}\|_2^2\right)$. We compare the results of MMSE-ADMM and MMSE-CADMM against the following reconstruction methods:

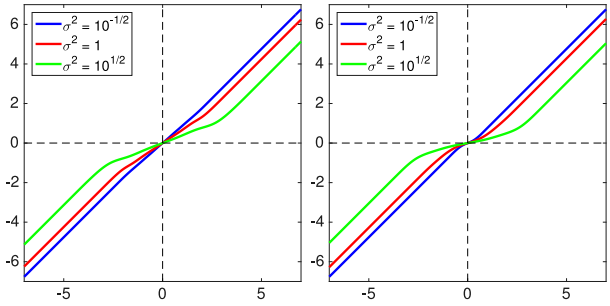
1) MMSE: This is the optimal estimator (in the MSE sense) and is obtained through a message-passing algorithm [39].
2) LMMSE (Linear MMSE): The best linear estimation is obtained by applying the Wiener filter to the noisy observation: $\hat{\boldsymbol{x}}_{\text{LMMSE}} = (\boldsymbol{I} + \sigma^2 \boldsymbol{L}^T \boldsymbol{L})^{-1}\boldsymbol{y}$. This is also the least-square solution with ℓ_2 (Tikhonov-like) regularization.
3) TV (Total Variation) [9]: This estimator is obtained with an ℓ_1 regularizer whose proximal operator is simply a soft-thresholding: $T_\lambda(u) = 1_{\{|u|>\lambda\}}\text{sign}(u)(|u| - \lambda)$. In our experiments, the regularization parameter λ is *optimized* for each signal.



(a) Brownian motion (b) Compound Poisson

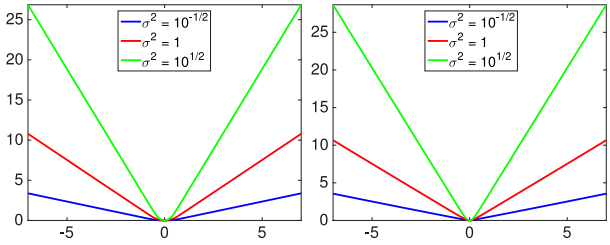
Fig. 4. Unconstrained shrinkage functions learned from data for various noise variances σ^2.

The denoising performances of MMSE-ADMM and MMSE-CADMM for various noise variances between $10^{-1/2}$ and $10^{1/2}$ are reported in Figs. 2 and 3, respectively. It is remarkable that both MMSE-ADMM and MMSE-CADMM curves are almost identical to the optimal MMSE curve (with the largest gap being about 0.1 dB) and significantly outperform TV, for both types of signals, and LMMSE, for compound Poisson processes. Note that, for Brownian motions, LMMSE and MMSE are the same. The unconstrained and constrained shrinkage functions learned for three different levels of noise are illustrated in Figs. 4 and 5, respectively. As can be seen in Fig. 4, the unconstrained learning might result in non-monotonic curves, which cannot be the proximal operators of any penalty functions, according to [32, Proposition 1]. By contrast, the antisymmetric and firmly nonexpansive curves in Fig. 5 are the proximal operators of the symmetric and convex penalty functions that are plotted in Fig. 6. These functions were numerically obtained by integrating $\partial\Phi = (T - \text{Id})^{-1}$, where T is the learned shrinkage function.



(a) Brownian motion (b) Compound Poisson

Fig. 5. Antisymmetric and firmly nonexpansive shrinkage functions learned from data for various noise variances σ^2.



(a) Brownian motion (b) Compound Poisson

Fig. 6. Symmetric and convex penalty functions that admit the learned constrained shrinkage functions in Fig. 5 as their proximal operators for various noise variances σ^2.

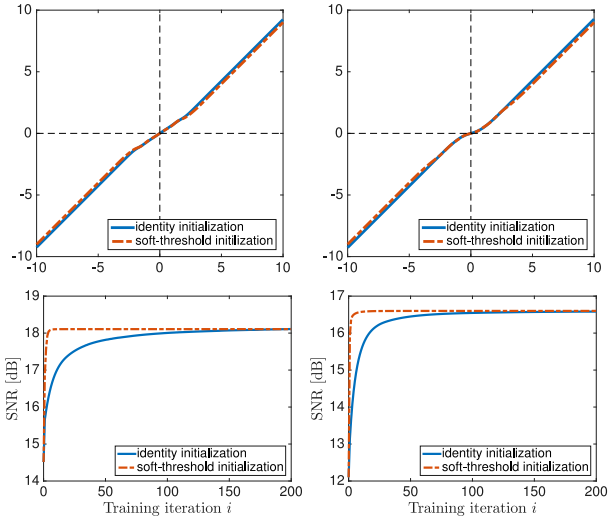
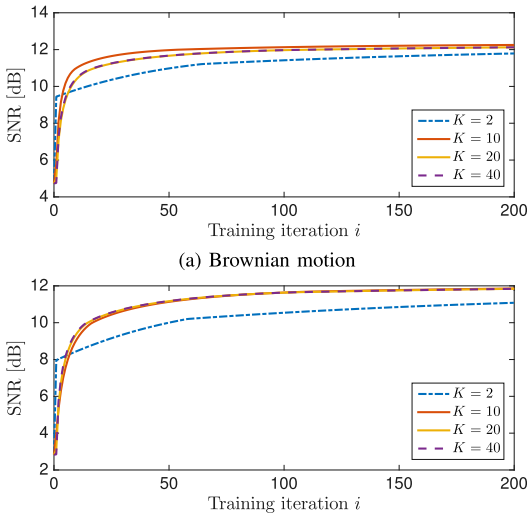


Fig. 7. First row: shrinkage functions learned via Algorithm 3 w.r.t. two different initializations for Brownian motion (left) and Compound Poisson (right). Second row: evolutions of the corresponding training SNRs. We used $\sigma^2 = 1$ and $K = 10$.

B. Stability of the Training

The constrained learning scheme is not only optimal in the testing phase but also very stable w.r.t. the initialization, $\boldsymbol{c}^{(0)}$, and the number of ADMM iterations, K, in the training phase.

Fig. 7 shows the evolutions of training SNRs together with learned shrinkage functions using Algorithm 3 with two different initializations: identity and soft-thresholding with parameter $\lambda = \sigma^2$. In this experiment, we fixed $\sigma^2 = 1$ and $K = 10$. It can be seen that the training cost functions of the two scenarios eventually converge to the same value and the resulting shrinkage functions are almost identical. Although training with the soft-thresholding initialization converges very quickly, we chose the identity initialization in all other experiments to demonstrate that our learning algorithms even work for such a blind initial guess.



(a) Brownian motion

(b) Compound Poisson

Fig. 8. Evolution of the training SNR using Algorithm 3 for different values of K (number of ADMM iterations) and for AWGN of variance $\sigma^2 = 10$.

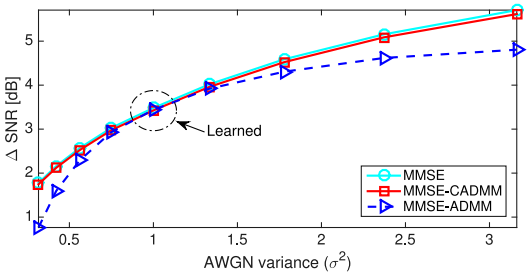
Fig. 8 illustrates the convergence of the training procedure using Algorithm 3 for varying number of ADMM iterations, K, in the extremely noisy case when $\sigma^2 = 10$. The learning rate γ is fixed for all choices of K. This experiment suggests that our backpropagation and gradient descent are not sensitive to the number of ADMM iterations, which can be interpreted as the number of layers in the underlying neural network [35]. The training SNR converges for all testing values of K, including the large ones. In principle, increasing K always results in a better SNR, but we experimentally observed that, when $K > 10$, the improvement is negligible for all levels of noise and for both type of testing signals. That is why we fixed $K = 10$ in all other experiments.

C. Constrained Versus Unconstrained Learning

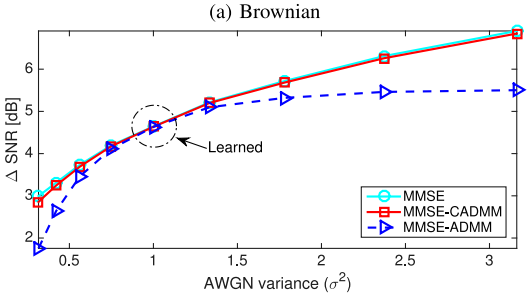
To demonstrate the benefits of the constrained learning over the unconstrained one, we compare their denoising performances for 9 different levels of noise as before, but this time only the shrinkage function T for $\sigma^2 = 1$ was learned. For constrained learning, the shrinkage function with respect to another noise variance σ^2 was numerically computed by using the formula

$$T_{\sigma^2} = \left(\sigma^2 T^{-1} + (1 - \sigma^2)\,\mathrm{Id}\right)^{-1}.$$

For unconstrained learning, these computations are prohibited, and so the learned shrinkage function for $\sigma^2 = 1$ was used for all the other noise levels. The results were illustrated in Fig. 9. It



(a) Brownian



(b) Compound Poisson

Fig. 9. Learning once and for all: only the shrinkage function for $\sigma^2 = 1$ is learned (with and without constraints) and the rest are obtained by scaling the learned penalty function with corresponding values of σ^2.

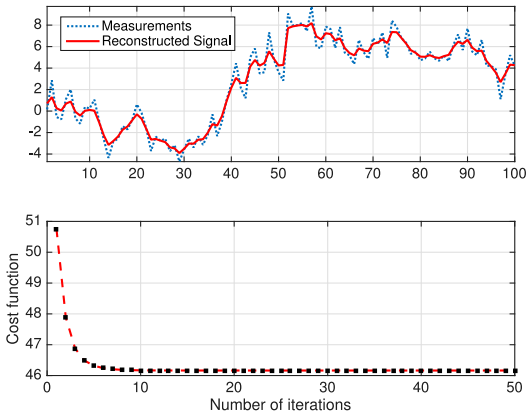


Fig. 10. Reconstruction of a specific Brownian motion with AWGN of variance $\sigma^2 = 1$ using MMSE-CADMM. Values of the underlying cost function are plotted for the first 50 iterations of ADMM.

is noticeable that MMSE-CADMM is much better than MMSE-ADMM and, surprisingly, almost as good as the optimal MMSE for all levels of noise, even though the (constrained) learning was performed only once. In other words, the experiments suggest that the proposed MMSE-CADMM combines desired properties of the MAP and MMSE estimators: fast implementation and scalability with noise variance of MAP and optimality of MMSE.

Another advantage of the constrained learning is its convergence guarantee that is associated with the minimization of an underlying cost function (as mentioned in Theorem 5), which does not necessarily exist in the case of unconstrained learning. Figs. 10 and 11 illustrates the reconstructions of a Brownian motion and a compound Poisson signal, respectively, from their

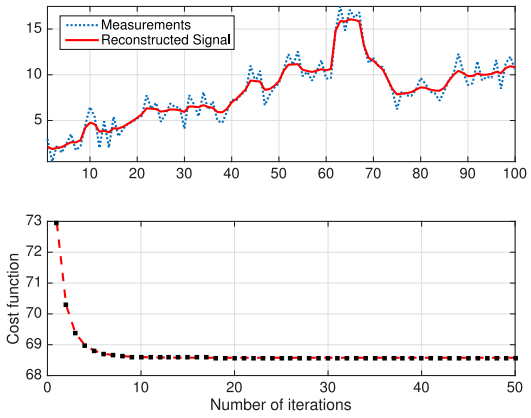


Fig. 11. Reconstruction of a specific compound Poisson signal with AWGN of variance $\sigma^2 = 1$ using MMSE-CADMM. Values of the underlying cost function are plotted for the first 50 iterations of ADMM.

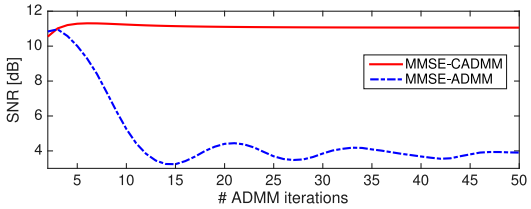


Fig. 12. Average SNRs when denoising compound Poisson signals with constrained and unconstrained learning schemes are plotted against the number of ADMM iterations used in the testing phase (K_{test}). The constrained and unconstrained shrinkage functions were both trained with $K_{\text{train}} = 2$ and $\sigma^2 = 10$.

noisy measurements using MMSE-CADMM, and the convergences of the corresponding cost functions. Experiments also show that the constrained learning is much more stable to the number of ADMM iterations used in the testing phase (K_{test}) when it is different from the number of ADMM iterations used in the training phase (K_{train}). Fig. 12 demonstrates this observation by plotting the average SNRs of denoising compound Poisson signals using MMSE-ADMM and MMSE-CADMM against K_{test} ranging from 2 to 50. In this experiment, both constrained and unconstrained learnings were performed with $K_{\text{train}} = 2$ and $\sigma^2 = 10$. It can be seen from the plot that, when K_{test} increases, the SNR of MMSE-ADMM tends to decrease and fluctuate significantly, while the SNR of MMSE-CADMM tends to improve and converge.

VII. Conclusion

We have developed in this paper a learning scheme for signal denoising using ADMM in which a single (iteration-independent) shrinkage function is constrained to be antisymmetric firmly-nonexpansive and learned from data via a simple projected gradient descent to minimize the reconstruction error. This constrained shrinkage function is proved to be the proximal operator of a symmetric convex penalty function. Imposing constraints on the shrinkage function gains several striking advantages: the antisymmetry reduces the number of learning parameters by a half, while the firm nonexpansiveness guarantees the convergence of ADMM, as well as the scalability with noise

level. Yet, the denoising performance of the proposed learning scheme is empirically identical to the optimal MMSE estimators for the two types of Lévy processes in a wide range of noise variances. Our experiments also demonstrate that learning the convex penalty function for one level of noise (via learning its proximal operator) and then scaling it for other noise levels yields equivalent performances to those of direct leaning for all noise levels. This property opens up an opportunity to vastly improve the robustness and generalization ability of learning schemes. In principle, The proposed learning method can be extended to the more general model $\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n}$, where $\boldsymbol{H}$ is a sampling matrix. In this paper, we have chosen to focus on the denoising model because its MMSE estimator is available for comparison. Potential directions for future research include extension of the proposed framework to general inverse problems as well as to multidimensional signals. Another issue worth investigating is the joint learning of the shrinkage function and the decorrelation (sparsifying) transform $\boldsymbol{L}$ from real data, like images, whose statistics are unknown.

APPENDIX
PROOF OF THEOREM 3

We first recall that

$$\operatorname{prox}_{\Phi} = (\partial\Phi + \mathrm{Id})^{-1}. \tag{22}$$

Assume for now that Φ is symmetric. Fix $\boldsymbol{x} \in \mathbb{R}^N$ and let $\boldsymbol{u} = \operatorname{prox}_{\Phi}(\boldsymbol{x})$, $\boldsymbol{v} = \operatorname{prox}_{\Phi}(-\boldsymbol{x})$. We need to show that $\boldsymbol{u} = -\boldsymbol{v}$. From (22), we have that

$$\begin{aligned} \boldsymbol{x} - \boldsymbol{u} &\in \partial\Phi(\boldsymbol{u}), \\ -\boldsymbol{x} - \boldsymbol{v} &\in \partial\Phi(\boldsymbol{v}). \end{aligned}$$

By the definition of the subdifferential operator, we obtain the following inequalities:

$$\Phi(-\boldsymbol{v}) - \Phi(\boldsymbol{u}) \geq \langle \boldsymbol{x} - \boldsymbol{u}\,,\, -\boldsymbol{v} - \boldsymbol{u} \rangle \tag{23}$$

$$\Phi(-\boldsymbol{u}) - \Phi(\boldsymbol{v}) \geq \langle -\boldsymbol{x} - \boldsymbol{v}\,,\, -\boldsymbol{u} - \boldsymbol{v} \rangle\,, \tag{24}$$

which, by the symmetry of Φ, can be further simplified to

$$\Phi(\boldsymbol{v}) - \Phi(\boldsymbol{u}) \geq \langle \boldsymbol{u} - \boldsymbol{x}\,,\, \boldsymbol{u} + \boldsymbol{v} \rangle \tag{25}$$

$$\Phi(\boldsymbol{u}) - \Phi(\boldsymbol{v}) \geq \langle \boldsymbol{x} + \boldsymbol{v}\,,\, \boldsymbol{u} + \boldsymbol{v} \rangle\,. \tag{26}$$

Adding these inequalities yields $\|\boldsymbol{u} + \boldsymbol{v}\|_2^2 \leq 0$, or $\boldsymbol{u} = -\boldsymbol{v}$.

Assume conversely that $\operatorname{prox}_{\Phi}$ is antisymmetric. We first show that $\boldsymbol{u} \in \partial\Phi(\boldsymbol{x})$ is equivalent to $-\boldsymbol{u} \in \partial\Phi(-\boldsymbol{x})$. Indeed, by using (22) and from the antisymmetry of $\operatorname{prox}_{\Phi}$,

$$\begin{aligned} \boldsymbol{u} \in \partial\Phi(\boldsymbol{x}) &\Leftrightarrow \boldsymbol{u} + \boldsymbol{x} \in \partial\Phi(\boldsymbol{x}) + \boldsymbol{x} = \operatorname{prox}_{\Phi}^{-1}(\boldsymbol{x}) \\ &\Leftrightarrow \boldsymbol{x} = \operatorname{prox}_{\Phi}(\boldsymbol{u} + \boldsymbol{x}) \\ &\Leftrightarrow -\boldsymbol{x} = \operatorname{prox}_{\Phi}(-\boldsymbol{u} - \boldsymbol{x}) \\ &\Leftrightarrow -\boldsymbol{u} - \boldsymbol{x} \in \operatorname{prox}_{\Phi}^{-1}(-\boldsymbol{x}) = \partial\Phi(-\boldsymbol{x}) - \boldsymbol{x} \\ &\Leftrightarrow -\boldsymbol{u} \in \partial\Phi(-\boldsymbol{x}). \end{aligned}$$

Furthermore, $\operatorname{prox}_{\Phi}(\boldsymbol{0}) = \boldsymbol{0}$ due to the antisymmetry. Since $\partial\Phi(\boldsymbol{0}) = \operatorname{prox}_{\Phi}^{-1}(\boldsymbol{0})$, it must be that $\boldsymbol{0} \in \partial\Phi(\boldsymbol{0})$. Let $G = \operatorname{gra}(\partial\Phi)$ and choose $(\boldsymbol{x}_0, \boldsymbol{u}_0) = (\boldsymbol{0}, \boldsymbol{0}) \in G$. Consider the Rockafellar anti-derivative [42] of $\partial\Phi$:

$$\begin{aligned} f(\boldsymbol{x}) &= \sup_{n \geq 1} \sup_{\substack{(\boldsymbol{x}_1, \boldsymbol{u}_1) \in G \\ \cdots \\ (\boldsymbol{x}_n, \boldsymbol{u}_n) \in G}} \left\{ \langle \boldsymbol{x} - \boldsymbol{x}_n\,,\, \boldsymbol{u}_n \rangle + \sum_{i=0}^{n-1} \langle \boldsymbol{x}_{i+1} - \boldsymbol{x}_i\,,\, \boldsymbol{u}_i \rangle \right\} \\ &= \sup_{n \geq 1} \sup_{\substack{(\boldsymbol{x}_1, \boldsymbol{u}_1) \in G \\ \cdots \\ (\boldsymbol{x}_n, \boldsymbol{u}_n) \in G}} \left\{ \langle \boldsymbol{x} - \boldsymbol{x}_n\,,\, \boldsymbol{u}_n \rangle + \sum_{i=1}^{n-1} \langle \boldsymbol{x}_{i+1} - \boldsymbol{x}_i\,,\, \boldsymbol{u}_i \rangle \right\} \end{aligned} \tag{27}$$

It is well known [38, Proposition 22.15] that $f \in \Gamma_0(\mathbb{R}^N)$ and $\partial f = \partial\Phi$. Therefore, we can invoke [38, Proposition 22.15] to deduce that $\Phi = f + c$, for some constant $c \in \mathbb{R}$. To show the symmetry of Φ, it suffices to show the symmetry of f. From (27) and by the symmetry of G, $f(-\boldsymbol{x})$ is equal to

$$\begin{aligned} &\sup_{n \geq 1} \sup_{\substack{(\boldsymbol{x}_1, \boldsymbol{u}_1) \in G \\ \cdots \\ (\boldsymbol{x}_n, \boldsymbol{u}_n) \in G}} \langle -\boldsymbol{x} - \boldsymbol{x}_n\,,\, \boldsymbol{u}_n \rangle + \sum_{i=1}^{n-1} \langle \boldsymbol{x}_{i+1} - \boldsymbol{x}_i\,,\, \boldsymbol{u}_i \rangle \\ &= \sup_{n \geq 1} \sup_{\substack{(\boldsymbol{x}_1, \boldsymbol{u}_1) \in G \\ \cdots \\ (\boldsymbol{x}_n, \boldsymbol{u}_n) \in G}} \langle -\boldsymbol{x} + \boldsymbol{x}_n\,,\, -\boldsymbol{u}_n \rangle + \sum_{i=1}^{n-1} \langle -\boldsymbol{x}_{i+1} + \boldsymbol{x}_i\,,\, -\boldsymbol{u}_i \rangle \\ &= \sup_{n \geq 1} \sup_{\substack{(\boldsymbol{x}_1, \boldsymbol{u}_1) \in G \\ \cdots \\ (\boldsymbol{x}_n, \boldsymbol{u}_n) \in G}} \langle \boldsymbol{x} - \boldsymbol{x}_n\,,\, \boldsymbol{u}_n \rangle + \sum_{i=1}^{n-1} \langle \boldsymbol{x}_{i+1} - \boldsymbol{x}_i\,,\, \boldsymbol{u}_i \rangle \\ &= f(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in \mathbb{R}^N, \end{aligned}$$

which shows that f is symmetric, completing the proof.

REFERENCES

[1] A. Tarantola, *Inverse Problem Theory and Methods for Model Parameter Estimation*. Philadelphia, PA, USA: SIAM, 2005.

[2] J. O. Ruanaidh and W. J. Fitzgerald, *Numerical Bayesian Methods Applied to Signal Processing*. New York, NY, USA: Springer, 2012.

[3] J. V. Candy, *Bayesian Signal Processing: Classical, Modern, and Particle Filtering Methods*. New York, NY, USA: Wiley, 2016.

[4] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," in *Proc. IEEE Global Conf. Signal Inf. Process.*, 2013, pp. 945–948.

[5] A. Rond, R. Giryes, and M. Elad, "Poisson inverse problems by the plug-and-play scheme," *J. Vis. Commun. Image Represent.*, vol. 41, pp. 96–108, 2016.

[6] S. H. Chan, X. Wang, and O. A. Elgendy, "Plug-and-play ADMM for image restoration: Fixed-point convergence and applications," *IEEE Trans. Comput. Imag.*, vol. 3, no. 1, pp. 84–98, Mar. 2017.

[7] S. Sreehari *et al.*, "Plug-and-play priors for bright field electron tomography and sparse interpolation," *IEEE Trans. Comput. Imag.*, vol. 2, no. 4, pp. 408–423, Dec. 2016.

[8] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1993.

[9] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, no. 1–4, pp. 259–268, 1992.

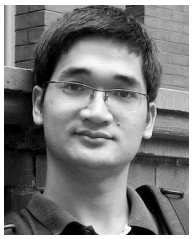
[10] C. Bouman and K. Sauer, "A generalized Gaussian image model for edge-preserving MAP estimation," *IEEE Trans. Image Process.*, vol. 2, no. 3, pp. 296–310, Jul. 1993.

[11] D. Krishnan and R. Fergus, "Fast image deconvolution using hyper-Laplacian priors," in *Proc. Adv. Neural Inf. Process. Syst. 23*, Vancouver, BC, Canada, Dec. 7–12, 2009, pp. 1033–1041.

[12] S. D. Babacan, R. Molina, and A. Katsaggelos, "Bayesian compressive sensing using Laplace priors," *IEEE Trans. Image Process.*, vol. 19, no. 1, pp. 53–64, Jan. 2010.

[13] M. A. T. Figueiredo and R. D. Nowak, "An EM algorithm for wavelet-based image restoration," *IEEE Trans. Image Process.*, vol. 12, no. 8, pp. 906–916, Aug. 2003.

[14] J. Bect, L. Blanc-Feraud, G. Aubert, and A. Chambolle, "A ℓ_1-unified variational framework for image restoration," in *Proc. Eur. Conf. Comput. Vis.*, 2004, pp. 1–13.
[15] I. Daubechies, M. Defrise, and C. D. Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Commun. Pure Appl. Math.*, vol. 57, no. 11, pp. 1413–1457, 2004.
[16] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.
[17] H. Choi and R. Baraniuk, "Wavelet statistical models and Besov spaces," in *Proc. SPIE Conf. Wavelet Appl. Signal Process.*, Denver CO, USA, 1999, pp. 489–501.
[18] M. Nikolova, "Model distortions in Bayesian MAP reconstruction," *Inverse Probl. Imag.*, vol. 1, no. 2, pp. 399–422, 2007.
[19] C. M. Stein, "Estimation of the mean of a multivariate normal distribution," *Ann. Statist.*, vol. 9, no. 6, pp. 1135–1151, 1981.
[20] R. Gribonval, V. Cevher, and M. E. Davies, "Compressible distributions for high-dimensional statistics," *IEEE Trans. Inf. Theory*, vol. 58, no. 8, pp. 5016–5034, Aug. 2012.
[21] M. Unser and P. D. Tafti, "Stochastic models for sparse and piecewise-smooth signals," *IEEE Trans. Signal Process.*, vol. 59, no. 3, pp. 989–1006, Mar. 2011.
[22] T. S. Cho, C. L. Zitnick, N. Joshi, S. B. Kang, R. Szeliski, and W. T. Freeman, "Image restoration by matching gradient distributions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 683–694, Apr. 2012.
[23] R. Gribonval, "Should penalized least squares regression be interpreted as maximum a posteriori estimation?" *IEEE Trans. Signal Process.*, vol. 59, no. 5, pp. 2405–2410, May 2011.
[24] M. Unser and P. D. Tafti, *An Introduction to Sparse Stochastic Processes*. Cambridge, U.K.: Cambridge Univ. Press, 2014.
[25] A. Amini, U. S. Kamilov, E. Bostan, and M. Unser, "Bayesian estimation for continuous-time sparse stochastic processes," *IEEE Trans. Signal Process.*, vol. 61, no. 4, pp. 907–920, Feb. 2013.
[26] E. Bostan, U. S. Kamilov, M. Nilchian, and M. Unser, "Sparse stochastic processes and discretization of linear inverse problems," *IEEE Trans. Image Process.*, vol. 22, no. 7, pp. 2699–2710, Jul. 2013.
[27] A. Kazerouni, U. S. Kamilov, E. Bostan, and M. Unser, "Bayesian denoising: From MAP to MMSE using consistent cycle spinning," *IEEE Signal Process. Lett.*, vol. 20, no. 3, pp. 249–252, Mar. 2013.
[28] P. Tohidi, E. Bostan, P. Pad, and M. Unser, "MMSE denoising of sparse and non-Gaussian AR(1) processes," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2016, pp. 4333–4337.
[29] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
[30] K. Gregor and Y. LeCun, "Learning fast approximation of sparse coding," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 399–406.
[31] U. S. Kamilov and H. Mansour, "Learning optimal nonlinearities for iterative thresholding algorithms," *IEEE Signal Process. Lett.*, vol. 23, no. 5, pp. 747–751, May 2016.
[32] U. Schmidt and S. Roth, "Shrinkage fields for effective image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 2774–2781.
[33] Y. Chen, W. Yu, and T. Pock, "On learning optimized reaction diffusion processes for effective image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 5261–5269.
[34] Y. Chen and T. Pock, "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1256–1272, Jun. 2017.
[35] Y. Yang, J. Sun, H. Li, and Z. Xu, "Deep ADMM-Net for compressive sensing MRI," in *Proc. Adv. Neural Inf. Process. Syst. 29*, 2016, pp. 1–9.
[36] S. Lefkimmiatis, "Non-local color image denoising with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3587–3596.
[37] K. G. G. Samuel and M. F. Tappen, "Learning optimized MAP estimates in continuously-valued MRF models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 477–484.
[38] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. New York, NY, USA: Springer, 2011.
[39] U. S. Kamilov, P. Pad, A. Amini, and M. Unser, "MMSE estimation of sparse Lévy processes," *IEEE Trans. Signal Process.*, vol. 61, no. 1, pp. 137–147, Jan. 2013.
[40] M. Unser, "Splines: A perfect fit for signal and image processing," *IEEE Signal Process. Mag.*, vol. 16, no. 6, pp. 22–38, Nov. 1999.
[41] C. Planiden and X. Wang, "Strongly convex functions, Moreau envelopes, and the generic nature of convex functions with strong minimizers," *SIAM J. Optim.*, vol. 26, no. 2, pp. 1341–1364, 2016.
[42] R. T. Rockafellar, *Convex Analysis*. Princeton, NJ, USA: Princeton Univ. Press, 1997.
[43] P. Thévenaz, T. Blu, and M. Unser, "Interpolation revisited," *IEEE Trans. Med. Imag.*, vol. 19, no. 7, pp. 739–758, Jul. 2000.



Ha Q. Nguyen was born in Hai Phong, Vietnam, in 1983. He received the B.S. degree in mathematics from the Hanoi National University of Education, Hanoi, Vietnam, the S.M. degree in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, MA, USA, and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign, Champaign, IL, USA, in 2005, 2009, and 2014, respectively.

During 2009–2011, he was a Lecturer in electrical engineering with the International University, Vietnam National University, Ho Chi Minh City, Vietnam, and during 2014–2017, he was a Postdoctoral Research Associate with Biomedical Imaging Group, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland. He is currently a Signal Processing Engineer with Viettel Research & Development Institute, Hanoi, Vietnam. His research interests include image processing, machine learning, computational imaging, data compression, and sampling theory.

Dr. Nguyen was a Fellow of Vietnam Education Foundation, cohort 2007. He was the recipient of the Best Student Paper Award (second prize) of the IEEE International Conference on Acoustics, Speech and Signal Processing in 2014 for his paper (with P. A. Chou and Y. Chen) on compression of human body sequences using graph wavelet filter banks.

Emrah Bostan (M'17) received the MSc. and PhD. degrees in electrical engineering from École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, in 2011 and 2016, respectively. He is currently a Postdoctoral Researcher with the Computational Imaging Lab, University of California, Berkeley, Berkeley, CA, USA. His research interest focuses on designing advanced signal/image processing algorithms for optical imaging applications.

Michael Unser (M'89–SM'94–F'99) is a full professor with École Polytechnique Fédéral de Lausanne, Lausanne, Switzerland. From 1985 to 1997, he was with Biomedical Engineering and Instrumentation Program, National Institutes of Health, Bethesda, MD, USA, conducting research on bioimaging. His primary area of investigation is biomedical image processing. He is internationally recognized for his research contributions to sampling theory, wavelets, the use of splines for image processing, stochastic processes, and computational bioimaging. He has authored or coauthored more than 300 journal papers on these topics. He is the author with P. Tafti of the book *An Introduction to Sparse Stochastic Processes* (Cambridge Univ. Press, 2014).

Dr. Unser is an EURASIP Fellow (2009) and a member of the Swiss Academy of Engineering Sciences. He was the Associate Editor-in-Chief (2003–2005) for the IEEE TRANSACTIONS ON MEDICAL IMAGING. He is currently a member of the editorial boards of *SIAM J. Imaging Sciences*, and *Foundations and Trends in Signal Processing.* He is the Founding Chair for the technical committee on Bio Imaging and Signal Processing of the IEEE Signal Processing Society. He was the recipient of several international prizes including three IEEE-SPS Best Paper Awards and two Technical Achievement Awards from the IEEE (2008 SPS and EMBS 2010).