Performance Analysis of Plug-and-Play ADMM: A Graph Signal Processing Perspective

Stanley H. Chan, Senior Member, IEEE

Abstract—The Plug-and-Play (PnP) ADMM algorithm is a powerful image restoration framework that allows advanced image denoising priors to be integrated into physical forward models to yield a provably convergent algorithm. However, despite the enormous applications and promising results, very little is known about why the PnP ADMM performs so well. This paper presents a formal analysis of the performance of PnP ADMM. Since its introduction in 2013 [1], the algorithm has demonstrated extremely promising results in image restoration and signal recovery problems [2]–[8]. However, despite the enormous applications of PnP ADMM and several studies on its convergence [8]–[10], it is generally unclear why the algorithm is performing so well. On the one hand, one can argue that the superior performance is attributed to the underlying image denoisers, which is evident from applications using convolutional neural networks [11]–[14]. Yet on the other hand, unless we can explicitly write down the optimization which describes the interaction of the denoisers, e.g., in the form of maximum-a-posteriori (MAP), it will be extremely difficult to quantify the solution even if we know the solution exists. The contribution of this paper is to present a set of formal performance analysis results for a family of denoisers known as the graph filters, so that we can understand the PnP ADMM behavior.

In order to state the problem more concretely, it would be useful to first review the PnP ADMM. The starting point of the PnP ADMM is the ADMM algorithm, which has become a standard tool for minimizing a sum of two separable functions of the form

$$\hat{x} = \arg\min_x f(x) + \lambda g(x).$$

(1)

Here, $f$ represents the objective function arising from the physical measurement models, e.g., blur, sampling, projection, or other linear transformations. The function $g$ is the regularization function, representing the prior model of the underlying signals, e.g., $\ell_p$-norm, sparsity, or total variation.

The parameter $\lambda$ is the regularization constant, and is assumed known a-priori and is fixed in this paper. Problems taking the form of (1) is broad, encompassing various linear inverse problems in imaging, deblurring, super-resolution, CT, MRI, and compressed sensing, to name a few.

ADMM solves (1) by converting the unconstrained optimization into a constrained problem

$$\hat{(x, v)} = \arg\min_{x, v} f(x) + \lambda g(v), \text{ subject to } x = v,$$

(2)

and considers the augmented Lagrangian function:

$$L(x, v, u) = f(x) + \lambda g(v) + u^T (x - v) + \frac{\rho}{2} \|x - v\|^2.$$\r

(3)

Then, the algorithm finds the solution by seeking a saddle point of $L$, which involves solving a sequence of subproblems in the form

$$x^{(k+1)} = \arg\min_{x \in \mathbb{R}^n} f(x) + \frac{\rho}{2} \|x - x^{(k)}\|^2,$$

(4)

$$v^{(k+1)} = \arg\min_{v \in \mathbb{R}^n} \lambda g(v) + \frac{\rho}{2} \|v - v^{(k)}\|^2,$$

(5)

$$\bar{u}^{(k+1)} = \bar{u}^{(k)} + (x^{(k+1)} - v^{(k+1)}),$$

(6)

where $\bar{u}^{(k)} \overset{\text{def}}{=} (1/\rho) u^{(k)}$ is the scaled Lagrange multiplier, $\hat{x}^{(k)} \overset{\text{def}}{=} v^{(k)} - \bar{u}^{(k)}$ and $\hat{v}^{(k)} \overset{\text{def}}{=} x^{(k+1)} + \bar{u}^{(k)}$ are the intermediate variables. Under mild conditions, e.g., when both $f$ and $g$ are closed, proper and convex, and if a saddle point of $L$ exists, one can show that the iterates returned by (4)-(6) converges to the solution of (1). Readers interested in knowing more about the theoretical properties of ADMM can consult tutorials such as [15].

The idea of PnP ADMM is to modify (5) by observing that it is essentially a denoising step, if we treat $\hat{v}^{(k)}$ as a "noisy" version of $v$ and $g(v)$ as a regularization for $v$. Based on this observation, we can replace (5) by a denoiser $D_\sigma : \mathbb{R}^n \rightarrow \mathbb{R}^n$ such that

$$u^{(k+1)} = D_\sigma (\hat{v}^{(k)}),$$

(7)

where $\sigma = \sqrt{N/\rho}$ is the denoising strength or a hypothesized "noise level". The choice of $D_\sigma$ is broad. $D_\sigma$ can be a proximal map such as total variation denoising, or an off-the-shelf image denoisers such as BM3D, non-local means, and more recently neural networks.

The author is with the School of Electrical and Computer Engineering, and the Department of Statistics, Purdue University, West Lafayette, IN 47907, USA. Email: stanchan@purdue.edu.
Replacing the explicit optimization in (5) by an off-the-shelf denoiser in (7) leads to many interesting questions:

- **Existence of g.** Given an arbitrary denoiser \( D_\sigma \), is it possible to have an equivalent \( g \) such that (5) and (7) coincide? If not, under what conditions of \( D_\sigma \) would this happen? Some recent studies provide an answer to this question [8], [16], which will be elaborated in later sections of this paper. An alternative approach is the Regularization by Denoising (RED) [17], which has advantages and limitations.

- **Convergence.** Does PnP ADMM converge? Clearly the convergence does not hold for arbitrary denoisers. For global convergence, the most basic requirement is that \( D_\sigma \) has symmetric gradient and is non-expansive [8]. For fixed point convergence, \( D_\sigma \) should be asymptotically invariant as \( \sigma \to 0 \) [9].

- **Performance.** If \( g \) does exist, what is it and why is it an effective regularization function? If \( g \) does not exists, then what does PnP ADMM solve, and why is it good? The focus of this paper is to address this performance issue. We will take two paths, one through the MAP formulation, and the other one through a concept called the consensus equilibrium.

- **Generalization.** How to generalize the algorithm to multiple denoisers? The generalization can be done using the consensus equilibrium technique. However, can we explain the performance, and what would be the optimal combination weights?

The main assumption we make in this paper is that the denoisers are in the family of symmetric smoothing filters [18]–[20] or graph filters [21]–[27]. These filters are simple in the sense that they are (pseudo) linear and hence they can be expressed using matrices. However, they are also representative enough to cover a sufficiently large group of image denoisers that we encounter in the literature.

The remaining of the paper is arranged as follows. We begin with a brief review of the construction of graph filters and their properties in Section 2. The main results are presented in Section 3 and 4. Section 3 analyzes the problem from a MAP perspective, whereas Section 4 approaches the problem from the consensus equilibrium perspective. Further discussions of open problems are presented in Section 5.

II. GRAPH FILTERS

A. Constructing a Graph Filter

Graph filters is a class of image denoisers mostly studied in the field of graph signal processing. The application of graph filters is very broad, e.g., deblurring using graph Laplacian [24], boosting image denoisers [22], image denoising using global similarity [25], JPEG compression using random walk [28], and blind deconvolution using reweighted graph total variation [27]. In terms of theory, there are studies of the graphs in continuous domain [26], sampling of graphs [29], and filter banks of graphs [30], to name a few.

Represented as matrices, graph filters take the form:

\[
\hat{x} = D_\sigma(y) = W y,
\]

where \( y \in \mathbb{R}^n \) is the noisy image, and \( \hat{x} \in \mathbb{R}^n \) is the denoised image. The matrix \( W \in \mathbb{R}^{n \times n} \) is the filter. There are multiple ways of constructing \( W \). One of the most commonly used approaches is to define a non-local weight kernel \( K \in \mathbb{R}^{n \times n} \):

\[
[K]_{ij} = \exp \left\{ -\frac{||y_i - y_j||^2}{2h^2} \right\},
\]

(9)

where \( y_i \in \mathbb{R}^d \) is a \( d \)-dimensional patch centered at pixel \( i \), and \( h \) is a parameter characterizing the denoising strength of the filter. Modifications to (9) are common, e.g., by incorporating spatial distance (which gives the non-local means filter), restricting to pixels instead of patches (which gives the bilateral filter), or extending the norm to a weighted norm (which gives the kernel regression filter). For more examples of the kernel matrices, we refer the readers to [18].

The \( K \) matrix defined in (9) is symmetric but the row sum \( K1 \) is not 1. Thus, \( K \) cannot be used directly as a denoising filter because it amplifies or attenuates signals. To normalize the matrix while preserving the symmetry, the most popular method is by means of the Sinkhorn-Knopp balancing algorithm [19], [20], [31], which iteratively normalizes the rows and columns of \( K \) until convergence. When \( K \) is symmetrized, the resulting matrix is called a symmetric smoothing filter, given by

\[
W = D^{-1/2} K D^{-1/2},
\]

(10)

where the diagonal matrix \( D \) is defined such that \( W1 = 1 \) and \( W^T 1 = 1 \). Since the columns and rows of \( W \) sum to 1, \( W \) is a doubly stochastic matrix.

B. Properties of Graph Filters

There are a few interesting properties of \( W \).

- \( W \) can be considered as an undirected graph. \( [W]_{ij} \) is the weight on the edge linking node \( i \) and node \( j \).

- \( 0 \leq \lambda_n(W) \leq \cdots \leq \lambda_1(W) = 1 \). This follows from the fact that \( W \) is doubly stochastic, and so the eigenvalues are all non-negative.

- \( I - W \) is the graph Laplacian, with the zero eigenvalue associated to the vector \( \frac{1}{\sqrt{n}}1 \).

- The regularization defined through the graph Laplacian \( x^T(I - W)x \) can be interpreted as

\[
x^T(I - W)x = \sum_{i=1}^{n} \sum_{j \neq i} [K]_{ij} \left( \frac{x_i}{\sqrt{[D]_{ii}}} - \frac{x_j}{\sqrt{[D]_{jj}}} \right)^2,
\]

(11)

which is measuring the smoothness of the signal \( x \) with the weights defined by the graph \( W \).

- The symmetric matrix \( W = D^{-1/2} K D^{-1/2} \) has a better denoising performance than its non-symmetric counter part \( W = \tilde{D}^{-1} K \) where \( \tilde{D} = \text{diag}(K1) \). This is attributed to the implicit clustering of the pixels during the Sinkhorn-Knopp balancing algorithm [20], among other reasons such as reduced degree of freedom so that the drop in variance overrides the gain in bias [19].

In order to present the theoretical results of this paper, we make an assumption about \( W \):
Assumption 1. We assume that the matrix $W$ is invertible, i.e., there exists a $W^{-1}$ such that

$$W^{-1}W = I, \quad \text{and} \quad WW^{-1} = I. \quad (12)$$

Assumption 1 is very strong and could be unrealistic for practical $W$’s. However, for the derivation of the main results, at least in Section 3, we need $W^{-1}$ to simplify some of the technical details. In Section IV when we discuss the performance from the consensus equilibrium perspective, we will lift the assumption.

Remark: Pseudo-Linearity. An important remark we should pay attention to is the pseudo-linearity of $W$ in the actual PhP ADMM implementation. That is, the matrix $W$ is a function of the input noisy image $y$, or more precisely $\hat{x} = W(y)y$ instead of (8). However, as mentioned in [32], the dependency of $W$ on $y$ is typically small if $W$ is estimated from a pre-filtered version of $y$, for example applying a baseline denoising algorithm to $y$ and constructing a new weight matrix $W$ based on the baseline estimate. Practically this is also justified in applications such as [8], where $W$ is updated for the first tens of iterations and is kept fixed for the remaining iterations. Following the same line of arguments, we assume that $W$ is pre-defined and is independent of $y$. Of particular interest is the oracle case where $W$ is estimated from the ground truth.

III. Performance Analysis via MAP

In this section we present the first set of analytic results. We call it the maximum-a-posteriori (MAP) analysis because we need to explicitly derive the MAP optimization.

A. Existence of $g$

Given a graph filter $D_\sigma : \mathbb{R}^n \rightarrow \mathbb{R}^n$ with $D_\sigma(x) = Wx$, the first question we should ask is whether it is possible to obtain a regularization function $g$ defined in (5). The rationale is that if we are able to do so, then we can characterize the MAP optimization in (1), and hence understand the performance.

Consider the minimization

$$\hat{x} = \text{argmin}_{x \in \mathbb{R}^n} g(x) + \frac{1}{2\sigma^2} \|x - \bar{x}\|^2, \quad (13)$$

the solution of which is characterized by

$$\hat{x} \in (I + \sigma^2 \partial g)^{-1} \bar{x}, \quad (14)$$

where $\partial g$ is the sub-differential of $g$, and the inverse operator is the resolvent of $g$ [15]. We assume that $D_\sigma$ is an invertible matrix $W \in \mathbb{R}^{n \times n}$. Therefore, $g$ must be continuously differentiable and so the subgradient at every point has a unique gradient. This replaces $\partial g$ by $\nabla g$, and substituting $\hat{x} = D_\sigma(\bar{x})$ into (14) yields

$$D_\sigma(\bar{x}) = (I + \sigma^2 \nabla g)^{-1}(\bar{x}), \quad \forall \bar{x}. \quad (15)$$

Define $G \in \mathbb{R}^{n \times n}$ as the matrix representation of $\nabla g$ such that $Gx = \nabla g(x)$. Since (15) holds for all $\bar{x}$, this further implies that the mappings are identical:

$$W = (I + \sigma^2 G)^{-1}. \quad (16)$$

Applying $I + \sigma^2 G$ to both sides yields $(I + \sigma^2 G) W = I$, and hence

$$G = \frac{1}{\sigma^2} (W^{-1} - I). \quad (17)$$

The identity in (17) offers a useful way of characterizing the existence of $g$.

Theorem 1 (Sreehari et al. [8] Theorem A.I, and Reehorst-Schniter [16] Theorem 1.). The mapping $g$ exists if and only if $G$ is symmetric.

Proof. The idea is to treat $g$ as a vector field which must be conservative. Therefore, the gradient $\nabla g$ must be symmetric using a classic result in vector calculus [33] (in particular Theorem 4.3.8 and 4.3.10).

For graph filters, the symmetry of $G$ is guaranteed because $W$ is symmetric. In addition, we also have that $W$ is non-expansive, as the eigenvalues of $W$ are bounded above by 1. Therefore, if we use a graph filter and if $W^{-1}$ exists, then according to [8] the corresponding minimization is a proximal map, and $g$ is convex, closed and proper. However, an arbitrary denoiser $D_\sigma$ does not necessarily have a symmetric Jacobian, and hence $g$ is not guaranteed to exist.

B. Deriving $g$

We can now state the first key result of this paper.

Theorem 2. Let $W \in \mathbb{R}^{n \times n}$ be a graph filter with full rank so that the inverse of $W$ exists. If $g : \mathbb{R}^n \rightarrow \mathbb{R}$ is defined by

$$g(x) = \frac{1}{2} x^T Gx = \frac{1}{2\sigma^2} x^T (W^{-1} - I)x, \quad (18)$$

then the solution of the minimization of (5) is

$$\hat{x} \overset{\text{def}}{=} \text{argmin}_{x \in \mathbb{R}^n} g(x) + \frac{1}{2\sigma^2} \|x - \bar{x}\|^2 = W \bar{x}. \quad (19)$$

Proof. The simplest proof is to substitute (18) into the optimization in (19). An alternative proof is to multiply both sides of (17) by $x$ to obtain

$$\nabla g(x) = Gx = \frac{1}{\sigma^2} (W^{-1} - I)x. \quad (20)$$

Then integrating both sides with respect to $x$ yields $g(x) = \frac{1}{2} x^T Gx$, which completes the proof.

Can we replace $W^{-1}$ by its pseudo-inverse $W^+$ if $W$ is not invertible? The answer is no. If we replace $W^{-1}$ by $W^+$ in (18), the optimization (13) becomes

$$\hat{x} = \text{argmin}_{x} \frac{1}{2\sigma^2} x^T (W^+ - I)x + \frac{1}{2\sigma^2} \|x - \bar{x}\|^2. \quad (21)$$

The first order optimality condition of (21) is given by $W^+ \hat{x} = \hat{x}$, which does not imply $\hat{x} = W \bar{x}$ because $W^+ W \neq I$ and $WW^+ \neq I$ unless $W$ has full rank. More specifically, $\hat{x} = W \bar{x}$ requires $\hat{x}$ to live in the range space of $W$. Since $W^+$ contains zero eigenvalues, there are multiple $\hat{x}$ which can be mapped to the same $\bar{x}$ through $W^+ \hat{x} = \bar{x}$. This is problematic, because the optimizer of (21) is no longer $\hat{x} = W \bar{x}$, and hence it violates our assumption that the denoiser is a graph filter.
C. The Role of $\rho$

Knowing the explicit expression of $g$, we can now analyze its performance. But before we proceed to the discussion, we should clarify the role of the parameter $\rho$. By substituting (18) into (2), and recalling $\sigma \equiv \sqrt{\lambda/\rho}$, we can show that the original problem which PnP ADMM solves is

$$\minimize_x f(x) + \frac{\rho}{2} x^T (W^{-1} - I)x.$$  

(22)

There is an alternative optimization in the graph signal processing literature known as the graph Laplacian, which we will study extensively in this paper:

$$\minimize_x f(x) + \frac{\lambda}{2} x^T (I - W)x.$$  

(23)

In (22), the original regularization parameter $\lambda$ is eliminated because $g$ contains its reciprocal. Thus, $\lambda$ is replaced by the ADMM internal parameter $\rho$. This internal parameter $\rho$ has no influence to the final solution in the classical ADMM because ADMM converges for all $\rho$ when $f$ and $g$ are closed, proper and convex. However, in PnP ADMM, the optimization cost is characterized by $\rho$. Thus, the solution would change as $\rho$ changes, a phenomenon reported in earlier papers, e.g., [9].

We can also ask: For the best $\rho$ and $\lambda$, how would the solutions of (22) and (23) behave? Figure 1 shows an experimental result where $f(x) = \frac{1}{2}\|x - y\|^2$. We vary $\rho$ (and equivalently $\lambda$) to evaluate the mean squared error (MSE) for a fixed oracle $W$. As shown in Figure 1, PnP ADMM and graph Laplacian have different operating regimes for $\rho$ and $\lambda$. PnP ADMM prefers smaller $\rho$, whereas graph Laplacian prefers larger $\lambda$. We will explain this observation in Section 4.

![Figure 1: Mean squared error of the solutions of (22) and (23) as $\rho$ (or $\lambda$) increases. PnP has a lower MSE than graph Laplacian for the best $\rho$ and $\lambda$.](image)

D. Comparison with Graph Laplacian

We can now compare the performance of PnP ADMM and the conventional graph Laplacian regularization. In order to do so, we consider a denoising problem such that

$$f(x) = \frac{1}{2}\|x - y\|^2,$$

where $y = x + \eta$, where $\eta \sim \mathcal{N}(0, \sigma^2_n I)$. That is, $y$ is a noisy version of $x$. Such choice of $f(x)$ is also justified by the $v$-subproblem (5), which is also a denoising problem.

We consider the following two optimizations:

$$\hat{x}_L = \arg\min_x \frac{1}{2}\|x - y\|^2 + \frac{\alpha}{2} x^T (I - W)x,$$  

(24)

$$\hat{x}_P = \arg\min_x \frac{1}{2}\|x - y\|^2 + \frac{\alpha}{2} x^T (W^{-1} - I)x.$$  

(25)

Here, the first optimization (24) uses graph Laplacian as the regularization, and the second optimization (25) uses the PnP regularization. We use a parameter $\alpha$ instead of $\rho$ or $\lambda$ to keep the notation simple. The solutions of the two optimizations are

$$\hat{x}_L = [(1 + \alpha)I - \alpha W^{-1}]^{-1} y \equiv A_L^{-1} y,$$  

(26)

$$\hat{x}_P = [(1 - \alpha)I + \alpha W^{-1}]^{-1} y \equiv A_P^{-1} y,$$  

(27)

where we defined matrices $A_L \equiv (1 + \alpha)I - \alpha W$ and $A_P \equiv (1 - \alpha)I + \alpha W$ to simplify notations.

Our goal is to analyze the mean squared error (MSE) of these two estimates. To this end, we first eigen-decompose $W$ as $W = USU^T$, where $S = \text{diag}(s_1, \ldots, s_n)$ is the eigenvalue matrix. Define $b \equiv U^T x$ as the projection of $x$ by columns of $U$. Then, the MSE of $\hat{x}_L$ and $\hat{x}_P$ can be found as follows.

**Proposition 1.** Define $\hat{x}_L$ and $\hat{x}_P$ via (26) and (27). Let $W = USU^T$, where $U$ is the eigenvector matrix, and $S = \text{diag}(s_1, \ldots, s_n)$ is the eigenvalue matrix. Define $b \equiv U^T x$. Then, the MSE of $\hat{x}_L$ and $\hat{x}_P$ are respectively

$$\text{MSE}_L \equiv \mathbb{E}\|\hat{x}_L - x\|^2 = \sum_{i=1}^n \frac{\alpha^2 (1 - s_i)^2 b_i^2 + \sigma^2_n}{(1 + \alpha - \alpha s_i)^2},$$  

(28)

$$\text{MSE}_P \equiv \mathbb{E}\|\hat{x}_P - x\|^2 = \sum_{i=1}^n \frac{\alpha^2 (1 - s_i)^2 b_i^2 + \sigma^2_n s_i^2}{(1 + (1 - \alpha) s_i)^2}.$$  

(29)

**Proof.** The MSE of $\hat{x}_L$ is

$$\text{MSE}_L = \mathbb{E}\|A_L^{-1}(x + \eta) - x\|^2$$  

$$= \|A_L^{-1} - I\| \|x\|^2 + \sigma^2_n \text{Tr}\left([A_L^{-1} A_L]^{-1}\right)$$  

$$= \left\|U \left(1 - \frac{1}{1 + \alpha - \alpha s_i} \right) U^T x\right\|^2 + \cdots$$  

$$+ \sigma^2_n \left\|U \left(1 - \frac{1}{1 + \alpha - \alpha s_i} \right) U^T \right\|^2\|x\|_F^2$$  

$$= \sum_{i=1}^n \left(1 - \frac{1}{1 + \alpha - \alpha s_i}\right)^2 b_i^2 + \sigma^2_n \left(1 - \frac{1}{1 + \alpha - \alpha s_i}\right)^2$$  

$$= \sum_{i=1}^n \frac{\alpha^2 (1 - s_i)^2 b_i^2 + \sigma^2_n}{(1 + \alpha - \alpha s_i)^2}.$$  

(31)

Similarly, the MSE of $\hat{x}_P$ is

$$\text{MSE}_P = \|A_P^{-1} - I\| \|x\|^2 + \sigma^2_n \text{Tr}\left([A_P^{-1} A_P]^{-1}\right)$$  

$$= \sum_{i=1}^n \frac{\alpha^2 (1 - s_i)^2 b_i^2 + \sigma^2_n s_i^2}{(1 + (1 - \alpha) s_i)^2}.$$  

(32)
Now, we can consider the $i$-th term of the sums:
\[
\begin{align*}
\text{MSE}_L^i &= \frac{\alpha^2(1 - s_i)^2b_i^2 + \sigma_n^2}{(1 + \alpha - \alpha s_i)^2}, \\
\text{MSE}_P^i &= \frac{\alpha^2(1 - s_i)^2b_i^2 + \sigma_n^2s_i^2}{(\alpha + (1 - \alpha)s_i)^2}.
\end{align*}
\] (33) (34)

It holds that for $s_i \to 0$,
\[
\begin{align*}
\lim_{s_i \to 0} \text{MSE}_L^i &= \frac{\alpha^2b_i^2 + \sigma_n^2}{(1 + \alpha)^2}, \quad (35) \\
\lim_{s_i \to 0} \text{MSE}_P^i &= b_i^2. \quad (36)
\end{align*}
\]

The limits in (35) and (36) provide important information about the performance. We make the following observations.

1. **Energy Concentration by $W$.** The graph filter $W$, when applied to a signal $x$, projects the signal onto the space spanned by its columns. The effectiveness of this projection is determined by the quality of $W$. In the oracle setting, $W$ ensures that most of the energy is captured by the first few eigenvectors. This is illustrated in the Figure 2, where we consider a 1D signal $x$ and its noisy version $y$. The weight matrix $W$ is constructed from $x$ using the non-local means kernel defined in (9). By projecting $x$ and $y$ using the eigenvector matrix $U$, we see that $U^T x$ has significantly fewer non-zeros than $U^T y$ for large eigen-indices. Since $U^T x$ is the vector $b$, we have the observation:

*Observation 1*: For large $i$, $b_i$ is typically very small.

![Fig. 2: Projecting a noisy signal $y$ and the ground truth signal $x$ by the principle eigenvectors. The two curves show $U^T y$ and $U^T x$, where $U$ is the eigenvector matrix of $W$.](image)

2. **Eigenvectors at High Frequency.** Based on Observation 1, an immediate conclusion from (35) and (36) is that when $b_i \to 0$, the two limits are respectively
\[
\begin{align*}
\lim_{b_i, s_i \to 0} \text{MSE}_L^i &= \frac{\sigma_n^2}{(1 + \alpha)^2}, \quad \text{and} \quad \lim_{b_i, s_i \to 0} \text{MSE}_P^i = 0.
\end{align*}
\]
In other words, the PnP regularization ensures that the MSE goes to zero for high frequency (i.e., large eigen-index $i$) whereas the graph Laplacian has residue MSE for high frequency. These can also be seen from the eigenvalues of $A_L^{-1}$ and $A_P^{-1}$:
\[
\begin{align*}
\lambda_i(A_L^{-1}) &= \frac{1}{1 + \alpha - \alpha s_i}, \quad \text{and} \quad \lambda_i(A_P^{-1}) = \frac{s_i}{(1 - \alpha)s_i + \alpha},
\end{align*}
\]
where $\lambda_i(\cdot)$ denotes the $i$-th eigenvalue of the matrix. If we plot $\lambda_i(A_L^{-1})$ and $\lambda_i(A_P^{-1})$ as functions of the eigen-index $i$, we will obtain a result shown in Figure 3. In this experiment we set $\sigma_n = 0.05$ and $\alpha = 0.2$. As $s_i \to 0$, we have
\[
\lim_{s_i \to 0} \lambda_i(A_L^{-1}) = \frac{1}{1 + \alpha}, \quad \text{and} \quad \lim_{s_i \to 0} \lambda_i(A_P^{-1}) = 0,
\]
which explain why $\text{MSE}_L^i$ reaches a steady state whereas $\text{MSE}_P^i$ reaches zero as $s_i \to 0$.

*Observation 2*: As $i$ increases so that $s_i \to 0$ and $b_i \to 0$, the eigenvalue $\lambda_i(A_L^{-1})$ approaches a non-zero constant, whereas $\lambda_i(A_P^{-1})$ approaches 0. The same observation applies to MSEs.

3. **Trade offs in Bias and Variance.** MSE is the sum of bias and variance. So we can decompose the MSE and inspect
\[
\begin{align*}
\text{Bias}_P &= \frac{\alpha^2(1 - s_i)^2b_i^2}{(\alpha + (1 - \alpha)s_i)^2}, \quad \text{and} \quad \text{Var}_P = \frac{\sigma_n^2s_i^2}{(\alpha + (1 - \alpha)s_i)^2} \\
\text{Bias}_L &= \frac{\alpha^2(1 - s_i)^2b_i^2}{(1 + \alpha - \alpha s_i)^2}, \quad \text{and} \quad \text{Var}_L = \frac{\sigma_n^2}{(1 + \alpha - \alpha s_i)^2}.
\end{align*}
\]

Since by construction $s_i$ is decreasing as $i$ grows, we can show that $\text{Var}_L^i \geq \text{Var}_P^i$.

**Proposition 2.** For any $0 \leq s_i \leq 1$, and $\alpha \leq 1$, it holds that $\text{Var}_L^i \geq \text{Var}_P^i$ for all $i$.

**Proof.** To prove this result we just need to show the function
\[
\begin{align*}
f(s_i) &= s_i^2(1 + \alpha - \alpha s_i)^2 - (\alpha + s_i - \alpha s_i)^2 \\
&= \alpha(s_i - 1)^2(\alpha s_i^2 - 2s_i - \alpha),
\end{align*}
\]
is non-positive. In this expression, the first two terms are non-negative. The last term $\alpha s_i^2 - 2s_i - \alpha$ is non-positive because $\alpha s_i^2 - 2s_i - \alpha \leq s_i - 2s_i - \alpha \leq 0$, where the first inequality holds since $\alpha s_i \leq 1$.

The conclusion of this proposition is that the variance of PnP is always lower than the variance of graph Laplacian.
Fig. 4: Bias and variance of PnP and graph Laplacian.

However, the bias of PnP is larger than the bias of graph Laplacian, as we show in Figure 4. More specifically, we can prove the following result:

**Proposition 3.** For any $0 \leq s_i \leq 1$, and $\alpha < 1/2$, it holds that $\text{Bias}_{\text{L}}^i \leq \text{Bias}_{\text{P}}^i$ for all $i$.

**Proof.** Since the numerators of $\text{Bias}_{\text{L}}^i$ and $\text{Bias}_{\text{P}}^i$ are identical, we only need to compare the denominator. Thus, we have

$$f(s_i) = (\alpha + (1 - \alpha)s_i)^2 - (1 + \alpha - \alpha s_i)^2$$

$$= (1 - s_i)[2(2\alpha + 1) - (2\alpha - 1)s_i] \leq 0$$

if $s_i \geq \frac{2\alpha + 1}{2\alpha - 1}$, which is true when $\alpha < 1/2$.

The reason why the overall MSE is lower for PnP is attributed to the observation below.

**Observation 3:** The drop in variance of PnP is more significant than the gain in bias, and hence PnP’s overall MSE is lower than that of graph Laplacian.

**4. When is Graph Laplacian Better?** The analysis above suggests conditions under which PnP will perform worse than graph Laplacian. This happens when the vector $b = U^T x$ is large, because then the denominator of $\text{Bias}_{\text{P}}^i$, will amplify more than that of $\text{Bias}_{\text{L}}^i$. This also suggests that PnP is more sensitive to the correctness of $W$ than graph Laplacian. That is, when constructing $W$ from a pre-filtered signal, PnP demands higher quality of the pre-filtered signal.

To confirm this claim, we conduct an experiment by constructing a pre-filtered signal via $\hat{x}_{\text{pre}} = x + \epsilon$, where $x \in \mathbb{R}^n$ is the true signal, and $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$. That is, the pre-filtered signal is a noisy version of the clean signal, with perturbation specified by the noise variance $\sigma^2$. We then construct a graph filter $W$ using $\hat{x}_{\text{pre}}$. Technically speaking, $W$ is now $W(\hat{x}_{\text{pre}})$. Our goal is to see how $\text{MSE}_{\text{L}}$ and $\text{MSE}_{\text{P}}$ would change when $\sigma$ increases.  

The result of this experiment is shown in Figure 5, where we observe that PnP has a considerably lower MSE than graph Laplacian when $\sigma$ is small. But as $\sigma$ increases, $\text{MSE}_{\text{P}}$ quickly increases and eventually goes above $\text{MSE}_{\text{L}}$. This explains the relatively weak robustness of PnP compared to graph Laplacian.

However, we should emphasize that such conclusion is only valid when $W$ is fixed. In practical PnP ADMM, the weight matrix $W$ is updated in every ADMM iteration. If $\sigma$ decreases as PnP iterates, then $W$ could become a good filter.

**Observation 4:** The performance of PnP could be worse than graph Laplacian when $W$ is estimated wrongly.

**5. Geometry: Truncating Eigenvectors of $x$.** If $W$ is oracle, is there any simple geometric interpretation, beside the bias-variance analysis, that can explain the performance?

By expressing the eigen-decomposition of the $W$ as

$$W = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} S_1 & 0 \\ 0 & S_2 \end{bmatrix} \begin{bmatrix} U_1^T & U_2^T \end{bmatrix},$$

(37)

the PnP ADMM regularization in (25) can be written as

$$x^T(W^{-1} - I)x$$

(38)

If the eigenvalues $S_2 = 0$, then the second term in (38) will dominate as $S_2^{-1}$ is infinity. In this case, we must have $U_2^T x = 0$ in order for the regularization to stay finite. Therefore, the overall optimization is now constrained:

$$\hat{x}_p = \arg\min_x \frac{1}{2} \|x - y\|^2 + \frac{\alpha}{2} x^T U_1(S_1^{-1} - I)U_1^T x$$

subject to $U_2^T x = 0$.

(39)

1 An alternative experiment would be to pick a specific baseline denoising algorithm and adjust its internal parameters. However, different baseline denoising algorithms have different characteristics and this could cause bias in the analysis. Thus, we choose to design a pre-filtered signal by adding iid noise.
The constraint provides a simple geometric interpretation. While the graph Laplacian regularization searches in the entire column space, PnP ADMM only searches for a subspace spanned by eigenvectors of non-zero eigenvalues. As we see in Observation 1, this provides a very strong denoising power when $W$ is the oracle. Referring to the decomposition in (37), the signal is spanned only by $U_1$ whereas the noise is spanned by both $U_1$ and $U_2$. By restricting $U_2^T x = 0$, we ensure that any noise living in $U_2$ will be eliminated. See Figure 6 for illustration. As a result, the denoising is significantly more effective than graph Laplacian.

![Image](image_url)

Fig. 6: PnP ADMM decomposes the optimization variable $x$ into two subspaces: $U_1$, the subspace encoding both signal and noise, and $U_2$, the subspace encoding only the noise. By restricting $x$ such that $U_2^T x = 0$, we force PnP ADMM to eliminate any noise living in $U_2$.

Observation 5: Geometrically, PnP ADMM restricts the search space so that the pure noise components are forced to zero.

E. Questions Not Answered

To summarize, the fundamental reason why PnP works better is that the graph filter $W$ has effectively removed the unwanted noise from $y$ before proceeding to additional processing. Thus, a stronger denoiser leads to better PnP ADMM algorithm.

One major limitation of the above MAP-based analysis is that it tries to match $D_\sigma$ with a proximal map associated with a regularization function $g$. This is a particularly strong requirement, because many denoisers are not symmetric and hence there is no corresponding proximal map. It also requires $W^{-1}$ to exist, which is difficult to have in practice, e.g., a uniform average operator does not have an inverse.

The question we ask is that if we are willing to give up the proximal map requirement, would it be possible to obtain another interpretation that can explain the performance for denoisers that do not have $W^{-1}$? This leads to the next section of equilibrium-based analysis.

IV. PERFORMANCE ANALYSIS VIA EQUILIBRIUM

The MAP optimization in (1) is not the only way to obtain $\hat{x}$. In [34], Buzzard et al. generalize PnP ADMM by a concept called consensus equilibrium, which allows us to characterize the solution of PnP ADMM by analyzing the equilibrium state of a dynamical system. In this section we discuss the PnP ADMM performance from such equilibrium perspective.

A. Equilibrium

The starting point of the equilibrium analysis is the PnP ADMM algorithm listed in (4)-(6). By defining a pair of functions $F$ and $G$ using (4) and (5) respectively, we can show that the PnP ADMM solution $\hat{x}$ satisfies

$$\hat{x} = F(\hat{x} + \hat{u}),$$

$$\hat{x} = G(\hat{x} - \hat{u}).$$

A solution $(\hat{x}, \hat{u})$ satisfying (40) and (41) is called a consensus equilibrium solution. To make this more specific to the readers, consider the following denoising example.

**Proposition 4.** Consider two operators $F$ and $G$:

$$F(x) = \text{argmin}_{z \in \mathbb{R}^n} \frac{1}{2} ||z - y||^2 + \alpha \frac{1}{2} ||z - x||^2,$$

$$G(x) = Wx$$

The consensus equilibrium solution $(\hat{x}, \hat{u})$ satisfies

$$((1 - \alpha)I + \alpha W^{-1}) \hat{x} = y$$

and $\hat{u} = (I - W^{-1}) \hat{x}$.

**Proof.** Since $G(x) = Wx$, it holds that $G(\hat{x} - \hat{u}) = W(\hat{x} - \hat{u})$, which implies that

$$W\hat{u} = W\hat{x} - \hat{x} \implies \hat{u} = (I - W^{-1})\hat{x}.$$ 

The optimality of $F$ gives

$$F(\hat{x} + \hat{u}) = (1 + \alpha)^{-1}(y + \alpha(\hat{x} + \hat{u}))$$

Substituting $\hat{u} = (I - W^{-1})\hat{x}$ into $F(\hat{x} + \hat{u}) = \hat{x}$ gives

$$(1 + \alpha)^{-1}(y + \alpha(\hat{x} + (I - W^{-1})\hat{x})) = \hat{x}.$$ 

Rearranging the terms yields the desired result.

The interesting fact about Proposition 4 is that, given $\hat{x}$ we can easily check whether it is a consensus equilibrium point. To illustrate this point, let us continue with the example in Proposition 4.

**Proposition 5.** Define the functions $\varphi$, $\psi_1$, $\psi_2$ and $\psi_3$:

$$\varphi(x) = \frac{1}{2} ||x - y||^2 + \frac{\alpha}{2} x^T(W^{-1} - I)x,$$

$$\psi_1(x) = \frac{1}{2} ||x - Wy||^2 - \frac{(1 - \alpha)}{2} x^T(I - W)x,$$

$$\psi_2(x) = \frac{1}{2} (x - y)^T W(x - y) + \frac{\alpha}{2} x^T(I - W)x,$$

$$\psi_3(x) = \frac{1}{2} ||x - [(1 - \alpha)I + \alpha W^{-1}]^{-1} y||^2.$$ 

The minimizers of these four functions all satisfy the consensus equilibrium condition (42).

**Proof.** Recalling (27), the minimizer of $\varphi$ is

$$\hat{x} = [(1 - \alpha)I + \alpha W^{-1}]^{-1} y,$$

which satisfies (42). Rearranging the terms we obtain

$$[(1 - \alpha)I + \alpha W^{-1}]^{-1} y = [(1 - \alpha)W + \alpha I]^{-1} W y = [I - (1 - \alpha)(I - W)]^{-1} W y,$$
which is the minimizer of $\psi_1$. The minimizers of $\psi_2$ and $\psi_3$ can be derived in a similar manner.

Proposition 5 shows four different optimization formulations that give the same consensus equilibrium solution. The first optimization in $\varphi$ is the MAP formulation we saw in Section 3. The other optimizations of $\psi_1$, $\psi_2$ and $\psi_3$ have some meaningful interpretations.

- **Interpretation of $\psi_1(x)$**. The optimization of $\psi_1$ is informative in the sense that it completely eliminates the inverse matrix $W^{-1}$, as opposed to the MAP in (25). The objective function $\frac{1}{2}\|x - W y\|^2$ says that instead of trying to minimize the residue between the optimization variable $x$ and the noisy observation $y$, we minimize $x$ with the filtered version $W y$. For oracle $W$'s where pure noise components are isolated at high frequencies (See Observation 5), $W y$ ensures that the pure noise components are eliminated. Therefore, when minimizing $\frac{1}{2}\|x - W y\|^2$, we are guaranteed to only search for the signal component.

The regularization term $-(1-\alpha)x^T(I - W)x$ is a concave function for any $0 \leq \alpha \leq 1$, because $I - W$ is positive semi-definite. The concave function injects high frequency components that are lost by $W y$. In the extreme cases when $\alpha = 1$, we have

$$\psi_1(x) = \frac{1}{2}\|x - W y\|^2,$$

and so the minimizer is just $\hat{x} = W y$. When $\alpha = 0$, we have

$$\psi_1(x) = \frac{1}{2}\|x - W y\|^2 - \frac{1}{2}x^T(I - W)x$$

$$= \frac{1}{2}\|x\|^2 - x^T W y + \frac{1}{2}\|W y\|^2 - \frac{1}{2}\|x\|^2 + \frac{1}{2}x^T W x$$

$$= \frac{1}{2}(x - y)^T W (x - y) + \frac{1}{2}y^T W (W - I)y,$$

and so the minimizer is $\hat{x} = y$. In order to control the amount of high frequency injection, a small $\alpha$ is more preferred than a large $\alpha$. This explains why in Figure 1 a smaller $\alpha$ works better for PnP ADMM.

- **Interpretation of $\psi_2(x)$**. The function $\psi_2$ offers an interpretation similar to $\psi_1$, e.g., the objective $(x - y)^T W (x - y)$ has no penalty over the subspace spanned by the eigenvectors of zero-eigenvalues, and so the regularization $x^T (I - W)x$ will force those dimensions to zero.

$\psi_2$ is also a special case of Kheradmand and Milanfar [24], who suggested

$$\psi(x) = \frac{1}{2}(x - y)^T (I + \beta(I - W))(x - y) + \frac{\alpha}{2}x^T (I - W)x$$

for any $\beta \geq -1$. If we let $\beta = -1$, then we obtain $\psi_2$.

- **Interpretation of $\psi_3(x)$**. The last example $\psi_3$, while still giving the consensus equilibrium solution, is more or less useless (in terms of interpretation) because it is just a restatement of the solution.

In summary, Proposition 5 shows that the solution of the PnP ADMM is completely characterized by the consensus equilibrium. Solving the MAP optimization is just one of the many possible ways of achieving the equilibrium. If we depart from the MAP formulation, i.e., if we are willing to change the forward modeling $f(x)$ by integrating the denoiser into the forward model, then there is a wide range of optimization formulations that can offer meaningful interpretations while satisfying the equilibrium condition.

### B. General Linear Inverse Problems

The equilibrium based approach also allows us to extend the analysis to general linear inverse problems which could be difficult to study by writing out the MSE in Section 3. The general linear inverse problem using the PnP ADMM with a graph filter requires two operators:

$$F(x) = \arg\min_{x \in \mathbb{R}^n} \frac{1}{2}\|Ax - y\|^2 + \frac{\alpha}{2}\|z - x\|^2,$$

$$G(x) = W x.$$

of which the equilibrium solution $\hat{x}$ satisfies

$$\left(A^T A + \alpha(W^{-1} - I)\right)\hat{x} = A^T y. \quad (48)$$

The condition in (48) can be written as

$$\left(W^\frac{1}{2} A^T A W^\frac{1}{2} + \alpha(I - W)\right)W^{-\frac{1}{2}} \hat{x} = W^{-\frac{1}{2}} A^T y,$$

which is also equivalent to

$$\hat{x} = W^{-\frac{1}{2}} \left(\arg\min_{z \in \mathbb{R}^n} \frac{1}{2}\|AW^{-\frac{1}{2}} z - y\|^2 + \frac{\alpha}{2}\|z^T (I - W) z\|\right). \quad (49)$$

While (49) may look unnatural at a first glance, in the sparse signal processing literature this is reminiscent to the MAP-synthesis minimization discussed by Elad [35].

How do we interpret (49)? The inner minimization is the standard graph Laplacian minimization with $Az$ changed to $AW^{-\frac{1}{2}} z$ in the objective. If $W$ has zero eigenvalues, then $W^{-\frac{1}{2}} z$ ensures that the columns along those directions are untouched. Then by the virtue of $z^T(I - W)z$, the minimizer must be zero along those directions because $I - W$ is positive semi-definite. The outer step by multiplying the solution with $W^{-\frac{1}{2}}$ ensures that the noise from the inner minimization is eliminated before creating the final solution.

### C. Performance of Multiple Priors

Beyond linear inverse problem we can extend the analysis to PnP ADMM involving multiple priors. In this case, the MAP-based optimization is

$$\hat{x} = \arg\min_{x \in \mathbb{R}^n} \varphi(x) \overset{\text{def}}{=} \mu_0 f_0(x) + \sum_{i=1}^k \mu_i f_i(x), \quad (50)$$

where $f_0 : \mathbb{R}^n \rightarrow \mathbb{R}$ is objective function associated with the forward model, and $\{f_i\}_{i=1}^k$ are the regularization functions associated with the prior models (if they exist).

Putting into the equilibrium framework, and if we restrict ourselves to graph filters, then we have

$$F_0(x) = \arg\min_{x \in \mathbb{R}^n} f_0(z) + \frac{\alpha}{2}\|z - x\|^2,$$

$$F_i(x) = W_i x, \quad i = 1, \ldots, k.$$
At equilibrium, the solution \((\hat{x}, \{\hat{u}_i\}_{i=1}^k)\) satisfies
\[
F_i(\hat{x} + \hat{u}_i) = \hat{x}, \quad \forall i \in \{1, \ldots, k\},
\]
which is a generalization of the case \(k = 2\) in (40) and (41). In (51), the mappings \(\{F_i\}_{i=1}^k\) are called agents, whose relative strengths are controlled by the parameters \(\{\mu_i\}_{i=1}^k\). Without loss of generality we assume that \(\sum_{i=1}^k \mu_i = 1\), and \(\mu_0\) can take any positive value. The intuition behind (51) is the concept of assembling individual experts to produce an overall better result. For example, \(F_i\) could represent image denoisers trained by different image classes, e.g., human faces, buildings and plants, etc. Given an unknown scene, it is likely that one of the denoisers would perform better than the others. Therefore, an ensemble of weak experts can improve the robustness against model mismatch.

To make the following discussion more concrete, let us focus on the denoising problem so that \(f_0(x) = \frac{1}{2}\|x - y\|^2\) for some noisy signal \(y\). Then, the agent \(F_0\) is defined as
\[
F_0(x) = \arg\min_x \frac{1}{2}\|z - y\|^2 + \frac{\alpha}{2}\|z - x\|^2 = \frac{y + \alpha x}{1 + \alpha}.
\]
The equilibrium condition can be determined by substituting \(x = \hat{x} + \hat{u}_0\), leading to
\[
\hat{x} = F_0(\hat{x} + \hat{u}_0) = \frac{y + (\alpha \hat{x} + \hat{u}_0)}{1 + \alpha},
\]
Rearranging the terms gives \(\hat{u}_0 = \frac{1}{\alpha}(\hat{x} - y)\). Combining with other agents, i.e.,
\[
\hat{x} = F_i(\hat{x} + \hat{u}_i) = W_i(\hat{x} + \hat{u}_i) \Rightarrow \hat{u}_i = (W_i^{-1} - I)\hat{x},
\]
we can substitute into \(\sum_{i=0}^k \mu_i \hat{u}_i = 0\) to show that the consensus equilibrium solution satisfies
\[
\left(\frac{\mu_0}{\alpha} I + \sum_{i=1}^k \mu_i (W_i^{-1} - I) \right) \hat{x} = \frac{\mu_0}{\alpha} y, \quad i = 1, \ldots, k.
\]
(52)

If we want to write (52) in terms of MAP, then a solution \(\hat{x}\) satisfying (52) is the minimizer of
\[
\hat{x} = \arg\min_{x \in \mathbb{R}^n} \frac{\mu_0}{\alpha} \frac{1}{2}\|x - y\|^2 + \frac{\alpha}{2} x^T \left(\sum_{i=1}^k \mu_i W_i^{-1} - I\right) x.
\]
(53)
The difference of (53) with the single-prior case, e.g., (43), is that we replace \(W^{-1}\) by a linear combination of multiple priors \(\sum_{i=1}^k \mu_i W_i^{-1}\).

To visually see the benefit of the combination, we consider \(k = 5\) filters \(\{W_i\}_{i=1}^k\) where the filters \(W_i\) are generated from kernels with different denoising strengths, see (9). The combination weights are set as \(\mu_i = 1/k\) for simplicity, and we set \(\mu_0 = 1\) and \(\alpha = 0.005\). The equilibrium condition of this combined filter gives a solution of the form
\[
\hat{x} = \left(\mu_0 I + \frac{1}{k} \sum_{i=1}^k \mu_i W_i^{-1} - I\right)^{-1} \mu_0 y.
\]
(54)

For comparison, we also consider solutions obtained from a single filter. That is, we use only one \(W_i\) with \(\mu_i = 1\). This leads to a solution
\[
\hat{x}_i = (\mu_0 I + \alpha (W_i^{-1} - I))^{-1} \mu_i y,
\]
for \(i = 1, \ldots, k\). The performance of the solutions are measured in terms of PSNR, and the PSNRs of this particular experiment are reported in Figure 7. Here, the legend \(W_i\) denotes the individual filters we use, and \(\hat{W}\) denotes the combined filter. In Figure 7 we also plot the eigenvalues of the respective inversion matrices. As one can see, the combined estimate staying in the middle \(W_3\) and \(W_4\), which are the two best performing matrices.

### D. Choosing the Weights

Is it possible to determine a set of weights \(\{\mu_i\}_{i=1}^k\) so that they are optimal in some sense? In principle this is doable by minimizing the mean squared error between the \(\hat{x}\) defined in (54) and the ground truth (if we have access to). However, such approach is computationally prohibited because the \(\{\mu_i\}_{i=1}^k\) are located in the denominator of (54).

A slightly more ad hoc approach is to solve (55) for every \(\hat{x}_i\), and define \(\hat{x}\) as a linear combination of them:
\[
\hat{x} = \sum_{i=1}^k \mu_i \hat{x}_i = \sum_{i=1}^k \mu_i A_i^{-1} y,
\]
(56)
where \(A_i \overset{\text{def}}{=} (I + (\alpha/\mu_0)(W_i^{-1} - I))\). An interpretation of (56) is the solution of an optimization
\[
\min_{\hat{x}, \{\hat{x}_i\}_{i=1}^k} \sum_{i=1}^k \mu_i \|\hat{x} - \hat{x}_i\|^2
\]
subject to \( (\mu_0 I + \alpha (W_i^{-1} - I)) \hat{x}_i = \frac{\mu_0}{\alpha} y\).
(57)

which says that among the individual equilibrium solutions \(\hat{x}_i\), we are finding an average \(\hat{x}\) that is close to all \(\hat{x}_i\)’s.

The weighted average in (56) provides a simple way of determining \(\{\mu_i\}_{i=1}^k\). Assuming that we have access to the ground truth \(x\), the weights \(\{\mu_i\}_{i=1}^k\) can be found by minimizing
\[
\min_{\{\hat{x}_i\}_{i=1}^k} \|\hat{x} - \sum_{i=1}^k \mu_i \hat{x}_i\|^2 \quad \text{subject to} \quad \sum_{i=1}^k \mu_i = 1.
\]
(58)
TABLE I: Inpainting Experiment. The input images contains of uniformly random sampled pixels at a rate of 80%, 60%, 40%, and 20%. The reconstruction methods are graph Laplacian (Lps) and PnP ADMM (PnP). The regularization parameter ρ and λ are optimized for every image and every configuration.

| Img   | 80%      | 60%      | 40%      | 20%      | 80%      | 60%      | 40%      | 20%      |
|-------|----------|----------|----------|----------|----------|----------|----------|----------|
|       | Lps PnP  | Lps PnP  | Lps PnP  | Lps PnP  | Lps PnP  | Lps PnP  | Lps PnP  | Lps PnP  |
| Barb  | 26.36    | 28.97    | 26.30    | 27.88    | 26.75    | 26.80    | 23.64    | 24.36    |
| Boat  | 26.64    | 28.54    | 25.93    | 27.31    | 25.16    | 25.96    | 23.04    | 23.46    |
| Cnra  | 26.39    | 27.88    | 25.30    | 26.33    | 24.13    | 24.81    | 22.01    | 22.26    |
| Coup  | 26.55    | 28.46    | 25.87    | 27.25    | 25.19    | 25.98    | 23.22    | 23.67    |
| Hous  | 27.28    | 30.52    | 27.22    | 29.83    | 26.95    | 28.67    | 25.07    | 25.82    |
| Lena  | 26.98    | 29.77    | 26.62    | 28.70    | 26.25    | 27.57    | 24.28    | 24.94    |
| Man   | 26.81    | 29.10    | 26.39    | 28.12    | 25.80    | 26.85    | 23.95    | 24.58    |
| Pepp  | 26.36    | 28.41    | 25.64    | 27.10    | 24.67    | 25.51    | 22.66    | 23.21    |

Letting $\mu = [\mu_1, \ldots, \mu_k]^T$, and defining the matrix $\Sigma \in \mathbb{R}^{k \times k}$ such that $[\Sigma]_{ij} = (x - \bar{x}_i)^T (x - \bar{x}_j)$, then (58) can be written as

$$\min_{\mu} \mu^T \Sigma \mu \quad \text{subject to} \quad \mu^T 1 = 1. \quad (59)$$

Closed-form solution of (59) exists, and is given by

$$\mu^* = \frac{\Sigma^{-1} 1}{1^T \Sigma^{-1} 1}. \quad (60)$$

Readers interested in the details can refer to [36] by Choi et al., where they discuss the solution’s geometry and its extension to cases where ground truth is not available.

V. FURTHER DISCUSSIONS

We close the paper by discussing a few issues that might be of interest to readers.

- **Experiment on Images.** Thus far the numerical results are all based on 1D signals. Readers who do image processing may wonder how well can the theory be translated to images. To articulate this problem, we consider an inpainting problem which has a forward model as

$$y = Sx + \eta. \quad (61)$$

In this equation, $S \in \mathbb{R}^{m \times n}$ is a sampling matrix which either selects or not selects a pixel from the input image $x$. The noise vector $\eta$ is assumed i.i.d. Gaussian with standard deviation $\sigma_\eta = 0.05$. Therefore, the objective function is $f(x) = \frac{1}{2} \| Sx - y \|^2$.

The result of this experiment is shown in Table I, where we compare graph Laplacian with PnP at different sampling ratios (20%, 40%, 60%, 80%). The $W$ matrix is the non-local means weight matrix symmetrized by the Sinkhorn-Knopp algorithm. For the estimated case, $\hat{W}$ is constructed from a pre-estimated image, where the pre-estimation is done using the Sheperd interpolation algorithm [37]. Tested on 8 “standard” images, the PSNR values are recorded and compared. The regularization parameters $\rho$ (or $\lambda$) are optimized for every image and for every testing condition. Therefore, the PSNRs shown in Table I are the best possible results of the methods. For both oracle and estimated $\hat{W}$, PnP ADMM is consistently better than graph Laplacian, which is coherent to the theoretical arguments.

- **Complexity Consideration.** When we compare (24) and (26), we tend to think that (24) is computationally less intensive because it is a standard least squares which can be solved using conjugate gradient (or other linear solvers). The PnP solution (26), on the other hand, requires spectral decomposition of $W^{-1}$ so that we can obtain the solution via, e.g., by solving an optimization (39). However, a complexity analysis based on this line of argument is not totally correct because in order to obtain the PnP solution we actually use the ADMM algorithm (4)-(6). Thus, the real complexity bottleneck is at the inversion, i.e., (4), not the graph filter $W$. For specific problems such as deblurring where $f(x)$ has structures, e.g., toeplitz matrix, the inversion could be done in $O(n \log n)$ via Fourier transform.

- **Statistical Interpretations of $W$ and $W^{-1}$.** If we regard $W$ as some kind of covariance matrix, then the inverse $W^{-1}$ is the precision matrix. The $(i, j)$-th element of the covariance matrix measures the correlation between node $i$ and node $j$ of the graph, whereas the $(i, j)$-th element of the precision matrix measures the partial correlation between node $i$ and node $j$ condition on all the other $n - 2$ nodes. Partial correlation is useful in handling situations where a high correlation between $i$ and $j$ is caused by some some third party node $k$. Thus by conditioning on $k$ we can remove the dependency of node $k$, and leaving just the partial correlation between $i$ and $j$.

As an example, in Figure 8 we show a ground truth signal $x$ used to generate a filter matrix $W$. We then compute the pseudo-inverse $W^+$ by keeping on the 50 leading eigenvectors, and plot its magnitude in the log-scale. (Note: We cannot compute the actual inverse $W^{-1}$ when $W$ is not invertible.) As we can see, while $W$ shows a banded diagonal behavior due to a spatial constraint we put, the inverse matrix $W^+$ shows a very strong signal dependent behavior.

- **Connection to RED.** A very similar idea to PnP ADMM is the regularization by denoising (RED) [17]. In RED, the idea is to define the regularization by

$$g(x) = \frac{1}{2} x^T (x - D_{\sigma}(x)). \quad (62)$$

One advantage of such formulation is that we are able to interpret the regularization as the denoiser $D_{\sigma}$ is explicitly expressed in $g$. However, as recently discussed by Reehorst and Schniter [16], while RED offers an interpretable regularization,
the existence of $g$ is problematic if $D_\sigma$ is not symmetric.

If we restrict $D_\sigma$ to symmetric smoothing filters so that $D_\sigma(x) = Wx$, then the RED regularization is nothing but

$$g(x) = \frac{1}{2} x^T (x - Wx) = \frac{1}{2} x^T (I - W)x,$$

which is exactly the graph Laplacian regularization. As we have shown in the analysis above, under the oracle setting, this does not have the noise rejection capability as PnP has.

- **Dynamics.** In this paper we assume that $W$ is defined through some pre-filtered signal, yet in practice $W$ is updated with the ADMM iterations. If $W$ is updated dynamically, obtaining a closed-form expression of the consensus equilibrium point would become very difficult. This is an open question, and more studies are needed to fully understand how to characterize the solution of this type.

- **Duality.** There is an interesting duality relationship between the graph Laplacian $(I - W)$ and the PnP prior $(W^{-1} - I)$. In fact, one can show that

$$\left( I - W \right)^{-1} - (W^{-1} - I)^{-1} = I. \quad (63)$$

This result is reminiscent to one of the recent theoretical studies by Feizi et al. [38] on generative adversarial network (GAN). The analogy is that if $(I - W)$ represents a discriminator for the real data, then $(I - W^{-1})$ represents a discriminator for the generated data. It would be interesting to derive similar duality interpretation from a graph perspective.

- **Beyond Image Restoration.** Thus far we have exclusively discussed imaging applications. If we consider $W$ as a forward diffusion, then $W^{-1}$ is the reversal diffusion. It would be interesting to investigate applications in general graph domain.

VI. CONCLUSION

We presented a formal performance analysis for the PnP ADMM algorithm. The analysis is based on the class of graph filters. Our findings suggest a few reasons why PnP ADMM performs better than the conventional graph Laplacian regularization. From the direct mean squared error (MSE) perspective, the inverse $W^{-1}$ associated with PnP ADMM ensures that the pure noise components are strongly attenuated during the ADMM algorithm, whereas they are less attenuated by the conventional graph Laplacian. As a result, the MSE of the PnP ADMM estimate has a much lower variance than the conventional graph Laplacian. PnP ADMM shows gain in bias, but the bias is typically smaller than the drop in variance unless the filter $W$ is poorly estimated.

To further understand the problem, this paper also introduces a performance analysis approach from an equilibrium perspective. Instead of deriving the solution from a MAP optimization, the equilibrium approach allows us to derive the solution by inspecting the equilibrium condition of the ADMM subproblems. This significantly alleviates the difficulty of assuming $W^{-1}$ to exist, tackling general linear inverse problems, and problems with multiple priors. We presented a number of equivalent optimization formulations and explained the behavior of the equilibrium solution.

As more sophisticated image restoration modules, e.g., deep neural networks, are being integrated into traditional model based recovery frameworks, we hypothesize that equilibrium will play an important role in analyzing the performance of these methods.

VII. ACKNOWLEDGEMENT

The author thanks Prof. Charles Bouman and Prof. Greg Buzzard of Purdue University for many inspiring discussions, Dr. Brentd Wohlberg of Los Alamos National Lab for offering valuable feedbacks on this paper, and Prof. Gene Cheung of York University for sharing thoughts on graph signal processing. This work is supported, in part, by the National Science Foundation under grants CCF-1763896 and CCF-1718007.

REFERENCES

[1] S. Venkatakrishnan, C. Bouman, and B. Wohlberg, “Plug-and-Play priors for model based reconstruction,” in Proc. IEEE Global Conference on Signal and Information Processing, 2013, pp. 945–948.
[2] Y. Dar, A. M. Bruckstein, M. Elad, and R. Giryes, “Postprocessing of compressed images via sequential denoising,” IEEE Trans. Image Process., vol. 25, no. 7, pp. 3044–3058, Jul. 2016.
[3] C. A. Metzler, A. Maleki, and R. G. Baraniuk, “From denoising to compressed sensing,” IEEE Trans. Information Theory, vol. 62, no. 9, pp. 5117–5144, Sep. 2016.
[4] A. Rond, R. Giryes, and M. Elad, “Poisson inverse problems by the Plug-and-Play scheme,” Journal of Visual Communication and Image Representation, vol. 41, pp. 96–108, Nov. 2016.
[5] U. S. Kamilov, H. Mansour, and B. Wohlberg, “A Plug-and-Play priors approach for solving nonlinear imaging inverse problems,” IEEE Signal Processing Letters, vol. 24, no. 12, pp. 1872–1876, Dec 2017.
[6] H. Chang and S. Marchesini, “A general framework for denoising phaseless diffraction measurements,” Available online: https://arxiv.org/abs/1611.01417.
[7] S. Ono, “Primal-dual Plug-and-Play image restoration,” IEEE Signal Processing Letters, vol. 24, no. 8, pp. 1108–1112, Aug 2017.
[8] S. Sreehari, S. V. Venkatakrishnan, B. Wohlberg, L. F. Drummy, J. P. Simmons, and C. A. Bouman, “Plug-and-Play priors for bright field electron tomography and sparse interpolation,” IEEE Trans. Computational Imaging, vol. 2, no. 4, pp. 408–423, Dec. 2016.
[9] S. H. Chan, X. Wang, and O. A. Elgendy, “Plug-and-Play ADMM for image restoration: Fixed point convergence and applications,” IEEE Trans. Computational Imaging, vol. 3, no. 5, pp. 84–98, Mar. 2017.
[10] A. M. Teodoro, J. M. Bioucas-Dias, and M. A. T. Figueiredo, “Scene-adapted Plug-and-Play algorithm with guaranteed convergence: Applications to data fusion in imaging,” Available online: https://arxiv.org/abs/1801.00605. 1

[11] F. Heide, M. Steinberger Y.-T. Tsai, M. Rouf, D. Pajkak, D. Reddy, O. Gallo, J. Liu, W. Heidrich, K. Egiazarian, J. Kautz, and K. Pulli, “FlexISP: A flexible camera image processing framework,” ACM Trans. Graph., vol. 33, no. 6, pp. 231:1–231:13, Nov. 2014. 1

[12] K. Zhang, W. Zuo, S. Gu, and L. Zhang, “Learning deep CNN denoiser prior for image restoration,” in Proc. IEEE Comput Soc Conf Comput Vis Pattern Recognit (CVPR’17), July 2017, pp. 2808–2817. 1

[13] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, “Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising,” IEEE Trans. Image Process., vol. 26, no. 7, pp. 3142–3155, July 2017. 1

[14] M. Borgerding, P. Schniter, and S. Rangan, “AMP-inspired deep networks for sparse linear inverse problems,” IEEE Trans. Signal Process., vol. 65, no. 16, pp. 4293–4308, Aug. 2017. 1

[15] 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, Jan. 2011. 1, 3

[16] E. T. Reehorst and P. Schniter, “Regularization by denoising: Clarifications and new interpretations,” Available online: https://arxiv.org/abs/1806.02296. 2, 3, 10

[17] Y. Romano, M. Elad, and P. Milanfar, “The little engine that could: Regularization by denoising (RED),” SIAM Journal on Imaging Sciences, vol. 10, no. 4, pp. 1804–1844, Oct. 2017. 2, 10

[18] P. Milanfar, “A tour of modern image filtering,” IEEE Signal Process. Mag., vol. 30, pp. 106–128, Jan. 2013. 2

[19] P. Milanfar, “Symmetrizing smoothing filters,” SIAM Journal on Imaging Sciences, vol. 6, no. 1, pp. 263–284, 2013. 2

[20] S. H. Chan, T. Zickler, and Y. M. Lu, “Understanding symmetric smoothing filters: A gaussian mixture model perspective,” IEEE Trans. Image Process., vol. 26, no. 11, pp. 5107–5121, Nov. 2017. 2

[21] F. Meyer and X. Shen, “Perturbation of the eigenvectors of the graph Laplacian: Application to image denoising,” Applied and Computational Harmonic Analysis, 2013, In press. Available online at http://arxiv.org/abs/1202.6666. 2

[22] H. Talebi, X. Zhu, and P. Milanfar, “How to SAIF-ly boost denoising performance,” IEEE Trans. Image Process., vol. 22, no. 4, pp. 1470–1485, Apr. 2013. 2

[23] K. M. Taylor and F. G. Meyer, “A random walk on image patches,” SIAM Journal on Imaging Sciences, vol. 5, pp. 688–725, 2012. 2

[24] A. Kheradmand and P. Milanfar, “A general framework for regularized similarity-based image restoration,” IEEE Trans. Image Process., vol. 23, no. 12, pp. 5136–5151, Dec. 2014. 2, 8

[25] H. Talebi and P. Milanfar, “Global image denoising,” IEEE Trans. Image Process., vol. 23, no. 2, pp. 755–768, Feb. 2014. 2

[26] J. Pang and G. Cheung, “Graph Laplacian regularization for image denoising: analysis in the continuous domain,” IEEE Trans. Image Process., vol. 26, no. 4, pp. 1770–1785, Apr. 2017. 2

[27] Y. Bai, G. Cheung, X. Liu, and W. Gao, “Graph-based blind image deblurring from a single photograph,” Available online: https://arxiv.org/abs/1802.07929. 2

[28] X. Liu, G. Cheung, X. Wu, and D. Zhao, “Random walk graph Laplacian-based smoothness prior for soft decoding of JPEG images,” IEEE Trans. Image Process., vol. 26, no. 2, pp. 509–524, Feb. 2017. 2

[29] A. Anis, A. Gadde, and A. Ortega, “Efficient sampling set selection for bandlimited graph signals using graph spectral proxies,” IEEE Trans. Signal Process., vol. 64, no. 14, pp. 3775–3789, Jul. 2016. 2

[30] N. Tremblay, P. Goncalves, and P. Borgnat, “Design of graph filters and filterbanks,” Available online at: https://arxiv.org/abs/1711.02046. 2

[31] R. Sinkhorn and P. Knopp, “Concerning non-negative matrices and doubly-stochastic matrices,” Pacific Journal of Mathematics, vol. 21, pp. 343 – 348, 1967. 2

[32] Y. Romano and M. Elad, “Boosting of image denoising algorithms,” SIAM Journal on Imaging Sciences, vol. 8, no. 2, pp. 1187–1219, 2015. 3

[33] S. Kantorovitz, Several real variables, Springer, 2016. 3

[34] G. T. Buzzard, S. H. Chan, S. Sreehari, and C. A. Bouman, “Plug-and-Play Unplugged: Optimization free reconstruction using consensus equilibrium,” Available online at: https://arxiv.org/abs/1705.08983, Dec. 2017. 7

[35] M. Elad, P. Milanfar, and R. Rubinstein, “Analysis versus synthesis in signal priors,” Inverse Problems, vol. 23, pp. 947–968, Apr. 2007. 8

[36] J. Choi, O. A. Elgendy, and S. H. Chan, “Optimal combination of image denoisers,” Available online: https://arxiv.org/pdf/1711.06712.pdf. 10

[37] D. Shepard, “A two-dimensional interpolation function for irregularly-spaced data,” in Proc. ACM National Conference, 1968, pp. 517–524. 10

[38] S. Feizi, C. Suh, F. Xia, and D. Tse, “Understanding GANs: the LQG Setting,” Available online: https://arxiv.org/abs/1710.10793. 11