Improving Pursuit Algorithms Using Stochastic Resonance

Dror Simon, Jeremias Sulam, Yaniv Romano, Yue M. Lu and Michael Elad

***Abstract*—Sparse Representation Theory is a sub-field of signal processing that has led to cutting edge results in many applications such as denoising, deblurring, super resolution and many other inverse problems. Broadly speaking, this field puts forward a model that assumes that signals are originated from a sparse representation in terms of an over-complete dictionary. Thus, when a corrupted measurement is given, we seek to estimate its original, clean form by finding the best matched sparse representation of the given signal in the dictionary domain. This process is essentially a non-linear estimation solved by a pursuit or a sparse coding algorithm.**

The concept of Stochastic Resonance (SR) refers to the counter-intuitive idea of improving algorithms' performance by a deliberate noise contamination. In this work we develop novel techniques that apply SR for enhancement of the performance of known pursuit algorithms. We show that these methods provide an effective MMSE approximation and are capable of doing so for high-dimensional problems, for which no alternative exists.

***Index Terms*—Sparseland, Stochastic Resonance, Basis Pursuit, Orthogonal Matching Pursuit, Noise-enhanced pursuit.**

I. Introduction

In signal processing, often times we have access to a corrupted signal and we wish to retrieve its clean version. This process includes a wide variety of problems, such as denoising, where we wish to remove noise from a noisy signal; deblurring where we look to sharpen an image that has been blurred or was taken out of focus; and inpainting in which we fill-in missing data that has been removed from the image. All the aforementioned tasks and many others, include a linear degradation operator and a stochastic corruption, and as such they can be described by the relation $\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{\nu}$, where $\boldsymbol{x}$ is the ideal signal, $\boldsymbol{H}$ is the linear degradation operator, $\boldsymbol{\nu}$ is the additive noise, and $\boldsymbol{y}$ stands for the noisy measurements.

In order to successfully restore $\boldsymbol{x}$, the Bayesian approach relies on some statistical properties of the corruption and a prior knowledge on the signal. When using a prior model assumption, we essentially estimate the original signal by obtaining a solution that is constrained by the model, while also being similar to the corrupted data. In image processing many such priors were developed over the years, among which one can mention the Total-Variation, exploiting self-similarity, relying on sparsity, and many others [1–3]. The literature in image processing shows an evolution of models, all seeking to improve the performance of the inverse problems described above. In this work we focus our attention to the sparse model prior, as described next.

The Sparseland model assumes that a signal $\boldsymbol{x} \in \mathbb{R}^n$ originates from an over complete dictionary $\boldsymbol{D} \in \mathbb{R}^{n\times m}$ where $n < m$, multiplying it by a representation vector $\boldsymbol{\alpha} \in \mathbb{R}^m$, i.e. $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\alpha}$. The vector $\boldsymbol{\alpha}$ is sparse, meaning that the number of non-zeros in it is very small compared to the data dimension,[1] $||\boldsymbol{\alpha}||_0 \ll n$. This implies that $\boldsymbol{x}$ is a linear combination of a small number of the columns from the dictionary $\boldsymbol{D}$, called *atoms*. One of the most fundamental problems in Sparseland is termed "Atom-Decomposition": Given $\boldsymbol{x}$, our goal is to find the sparsest explanation for it $\boldsymbol{\alpha}$. Essentially this calls for solving the (P_0) optimization problem:

$$(P_0): \quad \hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} ||\boldsymbol{\alpha}||_0 \quad \text{s.t.} \quad \boldsymbol{D}\boldsymbol{\alpha} = \boldsymbol{x}.$$

In typical situations, we do not get access to the clean signal $\boldsymbol{x}$, but rather to a corrupted version of it by some noise $\boldsymbol{\nu}$ with bounded energy $||\boldsymbol{\nu}||_2 \leq \epsilon$. The measurements are modeled as $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{\nu}$ and therefore the above (P_0) problem is modified in order to take the noise into account, leading to the following (P_0^ϵ) task:

$$(P_0^\epsilon): \quad \hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} ||\boldsymbol{\alpha}||_0 \quad \text{s.t.} \quad ||\boldsymbol{D}\boldsymbol{\alpha} - \boldsymbol{y}||_2 \leq \epsilon.$$

In the more general case, as described above, the degradation might include a corruption operator. In these cases, the resulting measurements are modeled as $\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{\nu}$, leading to a similar optimization problem in which the constraint is replaced by $||\boldsymbol{H}\boldsymbol{D}\boldsymbol{\alpha} - \boldsymbol{y}||_2 \leq \epsilon$. Once the problem is solved, we can estimate the original signal simply by $\hat{\boldsymbol{x}} = \boldsymbol{D}\hat{\boldsymbol{\alpha}}$.

The (P_0^ϵ) optimization is non-convex, posing a hard problem to tackle. Indeed, in general this problem is NP-Hard [4]. Nevertheless, approximation algorithms have been developed in order to manage this task effectively. One of these approximations, known as the Basis-Pursuit (BP) [5] is a relaxation method where the l_0 norm is replaced by an l_1. An alternative approach adopts a greedy strategy, such as in the case of the *Orthogonal Matching Pursuit* (OMP) [6], where the non-zeros in $\hat{\boldsymbol{\alpha}}$ are found one-by-one by minimizing the residual energy at each step.

These approximation algorithms have been accompanied by theoretical guarantees for finding a sparse representation $\hat{\boldsymbol{\alpha}}$ leading to a bounded error $||\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}||_2$ [7]. These results rely on the cardinality of $\boldsymbol{\alpha}$, the range of the non-zeros values and properties of the dictionary $\boldsymbol{D}$. In practice, under non-adversarial assumptions, these algorithms succeed with high

D. Simon, J. Sulam and M. Elad are with the Department of Computer Science of the Technion, Israel.

Y. Romano is with the Department of Electrical Engineering of the Technion, Israel.

Y. M. Lu is with the School of engineering and Applied Sciences, Harvard University, Cambridge, MA 02138, USA.

[1]The l_0 pseudo-norm stands for a count of the number of non-zeros in the vector.

probability even when the theoretical conditions of the worst case analysis are not met [8, 9].

Adopting a probabilistic point of view, as in [10], the fundamental purpose of these pursuit algorithms is to approximate the most likely support of the original signal, and therefore can be interpreted as a Maximum a Posteriori (MAP) estimator approximation under the sparsity prior. Clearly, the MAP is inferior to the Minimum Mean Square Error (MMSE) estimator in terms of mean square error and therefore it is not the optimal choice for some problems such as denoising. When applying a Bayesian approach, the MMSE is composed of an impractical sum of all the possible supports $S \in \Omega$

$$\hat{\boldsymbol{\alpha}}^{\text{MMSE}} = \mathbb{E}\left[\boldsymbol{\alpha}|\boldsymbol{y}\right] = \sum_{S\in\Omega} \mathbb{E}\left[\boldsymbol{\alpha}|\boldsymbol{y}, S\right] P(S|\boldsymbol{y})$$

and therefore it is usually avoided. As surprising as it might seem, the MMSE is actually a dense vector.

In a previous work [11] an MMSE estimator approximation under the Sparseland model was suggested. The proposed *Random-OMP* (RandOMP) algorithm has been proven to coincide with the MMSE estimator in the case where the dictionary $\boldsymbol{D}$ is unitary, and in the case where $\boldsymbol{D}$ is over-complete and $||\boldsymbol{\alpha}||_0 = 1$. RandOMP also improves the MSE results in general cases where the MMSE cannot be practically computed. In [12] a pursuit based on a Bayesian approach was suggested. The *Fast Bayesian Matching Pursuit* (FBMP) proposed a method to recover the most probable supports and approximate their posterior probabilities in order to achieve MMSE estimator approximation. Both algorithms, RandOMP and FBMP, rely on a greedy search where the support is updated one coefficient at a time. For this reason, these algorithms are restricted to low dimensional problems. In this work we propose methods to approximate the practically unattainable MMSE estimator regardless of the pursuit used, therefore being applicable to large dimensions as well.

The term *noise* has the natural connotation of unwanted disturbance in signal processing. Usually, this is indeed the case, and the scientific community often tries to diminish its influence in almost every signal processing application that involes estimation [13]. Without invalidating the previous statement, noise has also shown to be of great constructive value. Stochastic algorithms such as simulated annealing, genetic algorithms and image dithering rely on the properties of noise in order to succeed [5, 14, 15]. *Stochastic Resonance* (SR) is known as a phenomenon in which adding noise to a weak sub-threshold periodic signal increases its output Signal-to-Noise Ratio (SNR) when going through a threshold quantizer. This field has been further developed and has shown the ability to improve the performance of sub-optimal detectors [16], non-linear parametric estimators [1] and some image processing algorithms [17]. A well-known application that uses noise in order to improve a system's response is Dithering.

SR has been used in the past in order to improve sub-optimal non-linear systems' performance. As we saw throughout this section, pursuit algorithms form MAP estimator approximations. Could we consider a pursuit algorithm as a sub-optimal non-linear system whose performance can be improved by SR? In this work we intend to establish a novel MMSE approximation for the Sparseland model by integrating SR with known pursuit algorithms. More specifically, in our work we will seek answers to the following questions:

1) Can noise be employed to improve the performance of sparse coding algorithms?
2) How significant can this improvement be? Can we use noise enhancement to achieve MMSE approximation in the special unitary dictionary case?
3) Can we apply noise enhancement to approximate the MMSE of the denoising problem in a more general case?

In this paper we address these and closely related questions, showing how SR could be of great benefit for pursuit algorithms, and indeed provide MMSE approximation. Section II reviews Bayesian estimation in Sparseland and an initial intuition to SR in sparseland. In Section III we introduce SR pursuits in the special case where the dictionary is unitary and Section IV refers to the general dictionary case. Then, in In Section V we discuss two extensions regarding the SR and sparse model: one regarding the noise that should be used and another linking SR to Monte Carlo Methods. Section VI demonstrates a practical usage of our proposed algorithm for image reconstruction and image denoising and finally, in section VII we conclude this work.

II. Bayesian Estimation in Sparseland

Before we dive into the estimators themselves, we first introduce the generative model assumed. In this work we lean on the model introduced in [11] which is described as follows. $\boldsymbol{D} \in \mathbb{R}^{n\times m}$ is an over-complete dictionary $n < m$ and a sparse vector $\boldsymbol{\alpha} \in \mathbb{R}^m$ with either a known number of non-zeros $||\boldsymbol{\alpha}||_0 = K$ or some prior probability P_i for each atom to be chosen (we will variate between the two along this work). The non-zeros themselves, noted as $\boldsymbol{\alpha}_S$, are drawn from a Gaussian distribution $\boldsymbol{\alpha}_S \sim \mathcal{N}(0, \sigma_\alpha^2 \boldsymbol{I})$. Using the dictionary and sparse representation, we create a signal $\boldsymbol{x}$. The received samples are its noisy measurements $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{\nu} = \boldsymbol{D}\boldsymbol{\alpha} + \boldsymbol{\nu}$, where $\boldsymbol{\nu}$ is a random Gaussian noise, i.e. $\boldsymbol{\nu} \sim \mathcal{N}(0, \sigma_\nu^2 \boldsymbol{I})$. Unless stated otherwise, throughout the following sections we assume the described model and our goal is to acquire an estimation of the original signal $\hat{\boldsymbol{x}}(\boldsymbol{y})$. We shall now describe the estimators under the described model as were developed in [10].

The first estimator introduced is the Oracle estimator, which seeks to estimate the representation given the (oracle) information of it's support. The task of retrieving the true support S of the original sparse representation $\boldsymbol{\alpha}$ is the essence of the (P_0^ϵ) problem. If the support is known, then the MMSE estimator is simply the conditional expectation $\hat{\boldsymbol{\alpha}}_S^{\text{Oracle}} = \mathbb{E}\left[\boldsymbol{\alpha}|\boldsymbol{y}, S\right]$. From [10], this expectation has the following form:

$$\hat{\boldsymbol{\alpha}}_{S,\boldsymbol{y}}^{\text{Oracle}} = \frac{1}{\sigma_\nu^2} \boldsymbol{Q}_S^{-1} \boldsymbol{D}_S^T \boldsymbol{y}, \tag{1}$$

where $\boldsymbol{D}_S$ is a subdictionary containing only the columns (atoms) in the given support S, and $\boldsymbol{Q}_S$ is:

$$\boldsymbol{Q}_S = \frac{1}{\sigma_\alpha^2} \boldsymbol{I}_{|S|} + \frac{1}{\sigma_\nu^2} \boldsymbol{D}_S^T \boldsymbol{D}_S$$

originating from the Wiener filter solution. We refer to this estimator as the *Oracle* as there is no possible way of knowing the true support beforehand.

The second estimator is the MAP. In this case we look for the most probable support given our measurements and use it in order to estimate the signal[2].

$$\begin{aligned}\hat{S} &= \arg\max_S P(S|\boldsymbol{y}) = \arg\max_S P(\boldsymbol{y}|S)P(S) \\ &= \arg\max_S \frac{1}{2}\left|\left|\frac{1}{\sigma_v^2}\boldsymbol{Q}_S^{-\frac{1}{2}}\boldsymbol{D}_S^T\boldsymbol{y}\right|\right|_2^2 - \frac{1}{2}\log(\det(\boldsymbol{C}_S)) \\ &\qquad + \sum_{i\in S}\log(P_i) + \sum_{j\notin S}\log(1-P_j),\end{aligned}$$

where $\boldsymbol{C}_S^{-1} = \frac{1}{\sigma_\nu^2}\boldsymbol{I}_n - \frac{1}{\sigma_\nu^4}\boldsymbol{D}_S\boldsymbol{Q}_S^{-1}\boldsymbol{D}_S^T$ In the case where the number of non-zeros is known to be a constant $||\boldsymbol{\alpha}||_0 = K$ and all are equally likely, we can omit the last two sums since they indicate the prior of the support $P(S)$ and they are uniformly distributed. As described, the final estimator is:

$$\hat{\boldsymbol{\alpha}}_{\boldsymbol{y}}^{\text{MAP}} = \hat{\boldsymbol{\alpha}}_{\hat{S}^{\text{MAP}},\boldsymbol{y}}^{\text{Oracle}}$$

The last estimator is the MMSE. A well known result from estimation theory is that the MMSE estimator is given by the conditional expectation:

$$\begin{aligned}\hat{\boldsymbol{\alpha}}_{\boldsymbol{y}}^{\text{MMSE}} = \mathbb{E}\left[\boldsymbol{\alpha}|\boldsymbol{y}\right] &= \sum_S P(S|\boldsymbol{y})\mathbb{E}\left[\boldsymbol{\alpha}|\boldsymbol{y},S\right] \\ &= \sum_S P(S|\boldsymbol{y})\hat{\boldsymbol{\alpha}}_{S,\boldsymbol{y}}^{\text{Oracle}}.\end{aligned}$$

This is a weighted sum of all the possible supports. The probability $P(S|y)$ is given in [10]:

$$\begin{aligned}P(S|y) &= \frac{t_S}{t}, \qquad t \triangleq \sum_S t_S \\ t_s &\triangleq \frac{1}{\sqrt{\det(\boldsymbol{C}_S)}}\exp\left\{-\frac{1}{2}\boldsymbol{y}^T\boldsymbol{C}_S^{-1}\boldsymbol{y}\right\}\prod_{i\in S}P_i\prod_{i\notin S}(1-P_i).\end{aligned} \tag{2}$$

Note that both estimators, the MMSE and the MAP, are generally NP hard, since they require either to sum all the possible supports or to compute all the posterior probabilities and pick the highest one. Also, from the formulation given above, the essence of both estimators is the posterior probability $P(S|\boldsymbol{y})$. The better an algorithm estimates and leverages these probabilities, the better the original signal's estimation will be. All of the described estimators are for the representation vector $\boldsymbol{\alpha}$. In order to achieve estimations for the signal $\boldsymbol{x}$ we simply multiply $\hat{\boldsymbol{x}} = \boldsymbol{D}\hat{\boldsymbol{\alpha}}$ due to their linear relation. Given these estimators and the model we now begin our journey for pursuit improvement, starting with the unitary case.

As described in the Introduction, in this paper we use Stochastic Resonance in order to achieve MMSE approximation using many MAP approximations. The algorithm to do so is described in Algorithm 1. This algorithm simply adds a small amount of noise $\boldsymbol{n}_k$ to the already noisy signal $\boldsymbol{y}$ and then applies any pursuit given which is simply a MAP approximation and finally averages all the evaluations to a single final estimation. Note that two forms of estimations are given. The first one, the "non-subtractive", uses the extra-noisy evaluations as the basic estimators and averages them. The second one, the "subtractive", uses only the supports recovered by the noisy evaluations in order to achieve an estimation that is not effected by the added noise, and finally averages them out to achieve a final single estimation.

[2]Actually this is the MAP of the support. This is used in order to avoid the very probable case where the recovered signal is the **0** vector as described in [10]

Algorithm 1 SR Estimation algorithm

procedure SR-EST($\boldsymbol{y}$, $\boldsymbol{D}$, PursuitMethod, σ_n)
  for k=1...K **do**
    $\boldsymbol{n}_k \leftarrow$ SampleNoise(σ_n)
    $\tilde{\boldsymbol{\alpha}}_k \leftarrow$ PursuitMethod($\boldsymbol{y}+\boldsymbol{n}, \boldsymbol{D}$)
    $\hat{S}_k \leftarrow$ Support($\tilde{\boldsymbol{\alpha}}_k$)
    $\hat{\boldsymbol{\alpha}}_k \leftarrow \hat{\boldsymbol{\alpha}}_{\hat{S}_K}^{\text{Oracle}}(\boldsymbol{y})$
  end for
  $\hat{\boldsymbol{\alpha}}_{\text{non-subtractive}} = \frac{1}{K}\sum_{k=1}^K \tilde{\boldsymbol{\alpha}}_k$
  $\hat{\boldsymbol{\alpha}}_{\text{subtractive}} = \frac{1}{K}\sum_{k=1}^K \hat{\boldsymbol{\alpha}}_k$
  return $\hat{\boldsymbol{\alpha}}_{\text{non-subtractive}}, \hat{\boldsymbol{\alpha}}_{\text{subtractive}}$
end procedure

Before we dive in to the mathematical justifications given in the following sections, we first introduce the results of this algorithm using a simple synthetic example. In this experiment we generated random signals of length 50 with one non-zero (to make the MAP and MMSE attainable). Each non-zero is a Gaussian random variable, and the measurements are contaminated with additive white Gaussian noise $\boldsymbol{\nu}$. We used OMP as a basic MAP approximation and the reader can see the MSE results in Figure 1a. The improvement in performance is clearly seen. We repeated this experiment for a varying amount of added noise σ_v and tested the optimal SR MSE vs. the MMSE and the results can be seen in Figure 1b. Finally, in Figure 1c we show the optimal amount of SR added noise σ_n for different input noise energy values.

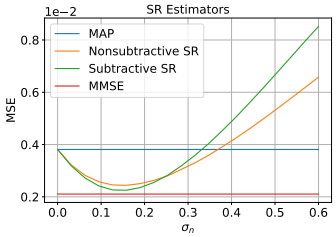
III. Improving Pursuits in the Unitary Case

A. The Unitary Sparse Estimators

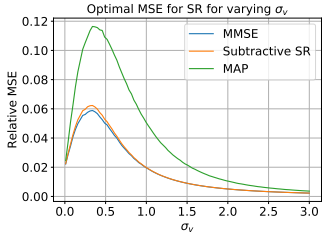
Continuing the Bayesian analysis of the sparse prior, we can specialize and simplify the expressions associated with the oracle estimator, MMSE and the MAP estimator. In [10] they mention the special case where the dictionary $\boldsymbol{D}$ is a unitary $n \times n$ matrix. In this case the dictionary is no longer over-complete and due to its special properties, the described estimators can be simplified. The MAP estimator is reduced to the elementwise Hard Thresholding operator applied on the projected measurements $\boldsymbol{\beta} = \boldsymbol{D}^T\boldsymbol{y}$, given by:

$$\hat{\alpha}_{MAP}(\beta) = \mathcal{H}_{\lambda_{MAP}}(\beta) = \begin{cases} c^2\beta & \text{if } |\beta| \geq \lambda_{MAP}, \\ 0 & else \end{cases},$$

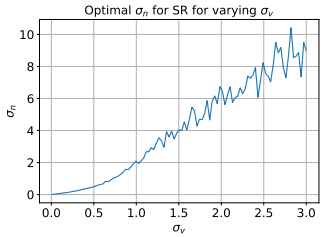
where we denote $c^2 \triangleq \frac{\sigma_\alpha^2}{\sigma_\alpha^2+\sigma_\nu^2}$ and $\lambda_{MAP} \triangleq \frac{\sqrt{2}\sigma_\nu}{c}\sqrt{\log\left(\frac{1-p_i}{p_i\sqrt{1-c^2}}\right)}$, and α and β are the elements of the vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$.



(a) MSE Comparison. σ_v=0.2



(b) Relative MSE is the MSE divided by the input noise σ_v^2 which represents a "denoising factor".



(c) Empirical optimal σ_n for varying σ_v values.

Fig. 1: 100 iterations of SR with OMP as a basic support estimator. $\boldsymbol{D}_{50\times100}$ is an over-complete normalized random dictionary. The measurements are $\boldsymbol{y} = \boldsymbol{D}\boldsymbol{\alpha} + \boldsymbol{\nu}$ where $\boldsymbol{\nu} \sim \mathcal{N}\left(\boldsymbol{0}, \sigma_{\boldsymbol{\nu}}^2 \boldsymbol{I}\right)$, $||\boldsymbol{\alpha}||_0 = 1$ and $\boldsymbol{\alpha}_s \sim \mathcal{N}(0,1)$.

The MMSE estimator in the unitary case is a simple elementwise shrinkage operator of the following form:

$$\hat{\alpha}_{MMSE} = \psi(\beta) = \frac{\exp\left(\frac{c^2}{2\sigma_\nu^2}\beta^2\right)\frac{p_i}{1-p_i}\sqrt{1-c^2}}{1+\exp\left(\frac{c^2}{2\sigma_\nu^2}\beta^2\right)\frac{p_i}{1-p_i}\sqrt{1-c^2}}c^2\beta.$$

Note that this shrinkage operator does not enforce a sparse vector, just as in the general case. The above scalar operators are also extended to act on vectors in an entry-wise manner.

B. The Unitary SR Estimators

In the previous section we saw that the MMSE is a weighted sum of all the probable solutions, where the weights are the posterior probabilities of each support. Similarly, we propose to approximate the MMSE by summing many probable solutions, by the following procedure: First, we add white zero mean Gaussian noise $\boldsymbol{n}_k$ to the signal $\boldsymbol{y}$. Note that since $\boldsymbol{D}$ is unitary, it does not matter if the noise is added to $\boldsymbol{y}$ or to its projection $\boldsymbol{\beta} = \boldsymbol{D}^T\boldsymbol{y}$. Then, we pass it through the MAP estimator $\mathcal{H}$ resulting with an estimation $\hat{\boldsymbol{\alpha}}_k$ of the original signal. This process is repeated many times, each time with a different noise realization. The final step includes an arithmetic mean over all the estimations[3]

$$\hat{\boldsymbol{\alpha}}_{stochastic} = \frac{1}{K}\sum_{k=1}^{K}\hat{\boldsymbol{\alpha}}_k = \frac{1}{K}\sum_{k=1}^{K}\mathcal{H}\left(\boldsymbol{\beta}+\boldsymbol{n}_k\right).$$

[3]Notice that the written equation operates on the the vectors element-wise.

The described process is an empirical arithmetic average approximating the expected value described as:

$$\begin{aligned}\mathbb{E}_n\left[\mathcal{H}_\lambda\left(\beta+n\right)\right] &= \int_{-\infty}^{\infty}\mathcal{H}_\lambda\left(\beta+n\right)p\left(n\right)dn \\ &= c^2\left[\beta Q\left(\frac{\lambda+\beta}{\sigma_n}\right)+\beta Q\left(\frac{\lambda-\beta}{\sigma_n}\right)\right]+ \\ &\qquad c^2\left[\frac{\sigma_n}{\sqrt{2\pi}}\left(e^{-\frac{(\lambda-\beta)^2}{2\sigma_n^2}}-e^{-\frac{(\lambda+\beta)^2}{2\sigma_n^2}}\right)\right],\end{aligned}$$

where we have denoted $Q(\bullet)$ as the tail probability of the standard normal distribution, i.e. $Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^\infty e^{-\frac{u^2}{2}}du$. The full derivation can be found in Appendix A. We shall define the above estimator as the *non-subtractive* SR estimator.

The term non-subtractive is used here to note that each estimation $\hat{\boldsymbol{\alpha}}_k$ is still contaminated with the noise $\boldsymbol{n}_k$ that was added in the process. Conversely, one might consider the *subtractive* estimator, in which we remove the projection of the added noise, resulting in the following shrinkage operator:

$$\begin{aligned}\hat{\alpha}_k\left(\beta, n_k\right) =& \mathcal{H}^-\left(\beta, n_k\right) \\ \mathcal{H}^-\left(\beta, n_k\right) =& \begin{cases} c^2\left(\beta+n_k\right)-c^2 n_k & \text{if } |\beta+n_k| \geq \lambda_{MAP}, \\ 0 & else \end{cases} \\ =& \begin{cases} c^2\beta & \text{if } |\beta+n_k| \geq \lambda_{MAP}, \\ 0 & else \end{cases}.\end{aligned}$$

Using this shrinkage operator and following the same process described above, we end up with the following estimator:

$$\begin{aligned}\mathbb{E}_n\left[\mathcal{H}^-\left(\beta,n\right)\right] =& \int_{-\infty}^{\infty}\mathcal{H}^-\left(\beta+n\right)p\left(n\right)dn \\ =& c^2\beta\left[Q\left(\frac{\lambda+\beta}{\sigma_n}\right)+Q\left(\frac{\lambda-\beta}{\sigma_n}\right)\right].\end{aligned}$$

Again, The full derivation can be found in Appendix A.

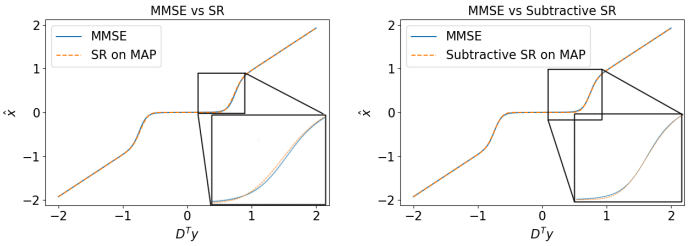
Notice that the described estimators have two parameters yet to be set: σ_n and λ. The former tunes the magnitude of the added noise, while latter controls the value of the thresholding operation. Note that the original MAP threshold might be sub-optimal due to the added noise and therefore, we leave λ as a parameter needed to be set. We will explore how to set these parameters later in this section.

C. Unitary SR Estimation Results

In order to demonstrate the similarity of the proposed estimators to the MMSE, we can simply compare their shrinkage curves, as seen in Figure 2. One can see that, while the curves do not overlap completely, for the right choice of parameters (λ and σ_n), the curves are indeed close to each other. In terms of MSE, in Figure 3 we can see the results of these methods as a function of σ_n (λ is fixed to the optimal value). It seems that the subtractive method is a slightly closer. We now discuss how to set the parameters in order to reach these optimal results.

D. Finding the Optimal Parameters for the Unitary Case

In the cases where the dictionary $\boldsymbol{D}$ is known but other parameters such as σ_α or σ_n are not known, the MMSE



(a) Non-subtractive SR (b) Subtractive SR

Fig. 2: The proposed SR estimators shrinkage curve compared to the MMSE. The parameters λ and σ_n values chosen to obtain the the optimal MSE.

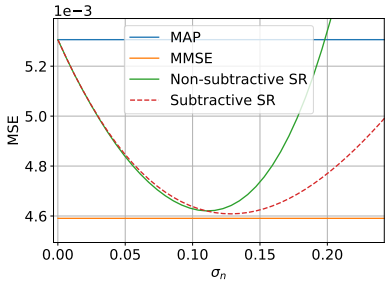


Fig. 3: 100 iterations of SR for varying σ_n values. $\boldsymbol{D}$ is a unitary 100×100 dictionary. The measurements are $\boldsymbol{y}=\boldsymbol{D\alpha}+\boldsymbol{\nu}$ where $\boldsymbol{\nu}\sim\mathcal{N}\left(\boldsymbol{0},\sigma_{\boldsymbol{\nu}}^2\boldsymbol{I}\right)$, $\sigma_{\boldsymbol{\nu}}=0.2$, $P_i=0.05\ \forall i$ and $\boldsymbol{\alpha}_s\sim\mathcal{N}\left(0,\boldsymbol{I}\right)$.

estimator cannot be computed. In such cases, we may try and estimate the MMSE by using SR. Since our added noise is Gaussian, we can use Stein's unbiased risk estimate (SURE) [18] which measures an estimator's MSE up to a constant, in order to optimize the free parameters λ and σ_n. The SURE formulation is:

$$\mu(\mathcal{H}(\boldsymbol{\beta},\lambda,\sigma_n),\boldsymbol{\beta})=||\mathcal{H}(\boldsymbol{\beta},\lambda,\sigma_n)||_2^2 - 2\mathcal{H}(\boldsymbol{\beta},\lambda,\sigma_n)^T\boldsymbol{\beta}+2\sigma_{\boldsymbol{\nu}}^2\nabla\mathcal{H}(\boldsymbol{\beta},\lambda,\sigma_n).$$

In the unitary case this is further simplified to an element-wise sum:

$$\begin{aligned}\mu(\mathcal{H}(\boldsymbol{\beta},\lambda,\sigma_n),\boldsymbol{\beta})&=\sum_i\mu(\mathcal{H}(\beta_i,\lambda,\sigma_n),\beta_i)\\&=\sum_i\mathcal{H}(\beta_i,\lambda,\sigma_n)^2-2\mathcal{H}(\beta_i,\lambda,\sigma_n)\beta_i\\&\quad+2\sigma_{\boldsymbol{\nu}}^2\frac{d}{d\beta_i}\mathcal{H}(\beta_i,\lambda,\sigma_n),\end{aligned}\quad(3)$$

and we wish to optimize σ_n and λ:

$$\sigma_n,\lambda=\arg\min_{\sigma_n,\lambda}\mu(\mathcal{H}(\boldsymbol{\beta},\lambda,\sigma_n),\boldsymbol{\beta}).$$

Plugging in the subtractive estimator $\mathcal{H}^-$ results in a closed form expression as can be seen in Appendix B. Also, in Appendix B we show the surface $\mathbb{E}_{\boldsymbol{n}}\mu$ for a specific experiment. Interestingly, we would like to note that empirically, the obtained optimal λ is quite close to the λ suggested by the MAP estimator.

IV. Improving Pursuits in the General Case

The unitary case is a good testing environment for new ideas since it is relatively simple and easy to analyze. That being said, SR in the unitary case has a slim importance since the MMSE has a closed form solution in the form of a shrinkage curve and therefore an approximation made by many sparse coded estimators is not needed. Also, most of the sparse theory applications use over-complete dictionaries.

In order to provide an analysis for the general case some assumptions should be made. We start by analyzing the single-atom case.

A. Single-Atom Analysis

In this section we analyze the over-complete case with the assumption that the cardinality of the sparse representation is known to be 1, i.e. $||\boldsymbol{\alpha}||_0=1$. From [11] we have that in this case, the MAP estimator described in the Introduction boils down to the following form:

$$\hat{\boldsymbol{x}}(\boldsymbol{y})=c^2\boldsymbol{y}_S,\qquad\boldsymbol{y}_S=\boldsymbol{d}_S\boldsymbol{d}_S^T\boldsymbol{y},\qquad c^2=\frac{\sigma_\alpha^2}{\sigma_\alpha^2+\sigma_\nu^2},$$

where the chosen atom $\boldsymbol{d}_S$ is:

$$\begin{aligned}\boldsymbol{d}_S&=\arg\min_{\boldsymbol{d}_S}||\boldsymbol{y}_S-\boldsymbol{y}||_2^2\\&=\arg\min_{\boldsymbol{d}_S}||\boldsymbol{d}_S\boldsymbol{d}_S^T\boldsymbol{y}-\boldsymbol{y}||_2^2=\arg\max_{d_S}\boldsymbol{y}^T\boldsymbol{d}_S\boldsymbol{d}_S^T\boldsymbol{y}.\end{aligned}\quad(4)$$

Following the subtractive concept we proposed in the unitary case, we introduce the following SR estimator:

$$\hat{\boldsymbol{x}}(\boldsymbol{y})=c^2\boldsymbol{y}_S,\qquad\boldsymbol{y}_S=\boldsymbol{d}_S\boldsymbol{d}_S^T\boldsymbol{y},$$

where this time the choice of the atom $\boldsymbol{d}_S$ is affected by an additional additive SR noise:

$$\boldsymbol{d}_S=\arg\min_{\boldsymbol{d}_S}||\boldsymbol{y}_S(\boldsymbol{n})-\boldsymbol{y}(\boldsymbol{n})||_2^2=\arg\max_{\boldsymbol{d}_S}(\boldsymbol{y}+\boldsymbol{n})^T\boldsymbol{d}_S\boldsymbol{d}_S^T(\boldsymbol{y}+\boldsymbol{n}).$$

Employing this as a pursuit to be used in Algorithm 1 we now analyze the proposed asymptotic estimator:

$$\mathbb{E}_{\boldsymbol{n}}\hat{\boldsymbol{x}}(\boldsymbol{y},\boldsymbol{n})=\mathbb{E}_{\boldsymbol{n}}c^2\boldsymbol{y}_S(\boldsymbol{n})=\mathbb{E}_S\left[\mathbb{E}_{\boldsymbol{n}|S}\left[c^2\boldsymbol{y}_S|S\right]\right]=c^2\sum_{i=1}^m\mathbb{E}_{\boldsymbol{n}|S}\left[\boldsymbol{d}_i\boldsymbol{d}_i^T\boldsymbol{y}\right]P\left(\hat{S}=i\right)=c^2\sum_{i=1}^m\boldsymbol{d}_i\boldsymbol{d}_i^T\boldsymbol{y}P\left(\hat{S}=i\right).$$

Similar to the MMSE in the general case (2), we have the sum of the solutions under all the possible supports (all supports with a single atom in this case), and each support is weighted by its probability to be chosen. We now turn to analyze this probability. As stated in Equation (4), the chosen atom i is the most correlated atom with the input signal:

$$\begin{aligned}P(\hat{S}=i)&=P\left(\left|\boldsymbol{d}_i^T(\boldsymbol{y}+\boldsymbol{n})\right|>\max_{j\neq i}\left|\boldsymbol{d}_j^T(\boldsymbol{y}+\boldsymbol{n})\right|\right)=\\&=P\left(|\tilde{n}_i|>\max_{j\neq i}|\tilde{n}_j|\right),\end{aligned}\quad(5)$$

where we defined $\tilde{\boldsymbol{n}}$ as a random Gaussian vector such that:

$$\tilde{\boldsymbol{n}} = \begin{bmatrix} \tilde{n}_1 \\ \tilde{n}_2 \\ \vdots \\ \tilde{n}_m \end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{D}^T\boldsymbol{y}, \sigma_n^2 \boldsymbol{D}^T\boldsymbol{D}\right).$$

We have that the probability of choosing the atom i is distributed as the probability of the maximum value of a random Gaussian vector with **correlated** variables. The vector's variables are correlated since in the non-unitary case, $\boldsymbol{D}^T\boldsymbol{D}$ is not a diagonal matrix.

Facing this dilemma, we can tackle it in several directions:

1) Instead of adding the Gaussian noise to the image $\boldsymbol{y}$, we can add it to the projected signal $\boldsymbol{D}^T\boldsymbol{y}$ thus avoiding the variables $\{\tilde{n}_i\}_{i=1}^m$ being correlated.
2) We can add some assumptions on the prior properties of the dictionary $\boldsymbol{D}$, thus deriving average case conclusions.
3) We can change the pursuit to use a constant threshold λ for the choice of the support instead of comparing the correlated variables. This means that when applying the algorithm, a cardinality of $|s| = 0$ or $|s| \geq 2$ might enter the averaging process. This is clearly a sub-optimal choice and we leave the study of this option for future work.

We shall now analyze the first two proposed alternatives.

1) Adding Noise to the Representation: Under this assumption, we continue from Equation (5), only this time the noise $\tilde{n}_i$ is white and has the following properties:

$$\tilde{\boldsymbol{n}} \sim \mathcal{N}\left(\boldsymbol{D}^T\boldsymbol{y}, \sigma_n^2 \boldsymbol{I}_{m\times m}\right)$$

Plugging this into (5):

$$\begin{aligned} P(\hat{S}=i) &= P\left(\left|\boldsymbol{d}_i^T\boldsymbol{y}+\boldsymbol{n}\right| > \max_{j\neq i}\left|\boldsymbol{d}_j^T\boldsymbol{y}+\boldsymbol{n}\right|\right) \\ &= P\left(|\tilde{n}_i| > \max_{j\neq i}|\tilde{n}_j|\right) \\ &= \int_0^\infty P\left(\max_{j\neq i}|\tilde{n}_j| < t \middle| |\tilde{n}_i| = t\right) P\left(|\tilde{n}_i| = t\right) dt \\ &= \int_0^\infty P\left(\max_{j\neq i}|\tilde{n}_j| < t\right) P\left(|\tilde{n}_i| = t\right) dt. \end{aligned} \quad (6)$$

Starting with the first element in the above product, since these are independent variables, we have:

$$\begin{aligned} &P\left(\max_{j\neq i}|\tilde{n}_j| < t\right) \\ &\quad = \prod_{j\neq i} P\left(|\tilde{n}_j| < t\right) = \prod_{j\neq i}\left[1 - P\left(|\tilde{n}_j| > t\right)\right] \\ &\quad = \prod_{j\neq i}\left[1 - \left(Q\left(\frac{t+\beta_j}{\sigma_n}\right) + Q\left(\frac{t-\beta_j}{\sigma_n}\right)\right)\right], \end{aligned}$$

where the last equality follows similar steps as in Appendix A with $\beta_i \triangleq \boldsymbol{d}_i^T\boldsymbol{y}$ and $t = \lambda$. The second term in (6) is simply an absolute value of a Gaussian, therefore:

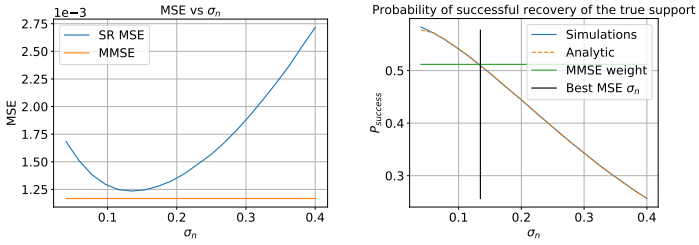
$$P\left(|\tilde{n}_i| = t\right) = \frac{1}{\sqrt{2\pi}\sigma_n}\left(e^{-\frac{(t-\beta_i)^2}{2\sigma_n^2}} + e^{-\frac{(t+\beta_i)^2}{2\sigma_n^2}}\right)$$

Putting the two terms back into (6):

$$\begin{aligned} P(\hat{s}=i) = \int_0^\infty \frac{1}{\sqrt{2\pi}\sigma_n}\left[e^{-\frac{(t+\beta_i)^2}{2\sigma_n^2}} + e^{-\frac{(t-\beta_i)^2}{2\sigma_n^2}}\right] \cdot \\ \prod_{j\neq i}\left[1 - \left(Q\left(\frac{t-\beta_j}{\sigma_n}\right) + Q\left(\frac{t+\beta_j}{\sigma_n}\right)\right)\right] dt \end{aligned} \quad (7)$$

The obtained expression cannot be solved analytically but can be numerically calculated. In Figure 4 we present a simulation for the single atom case where the noise is added to the projection $\beta_i = \boldsymbol{d}_i^T\boldsymbol{y}$. In Figure 4a we can observe that indeed, even in the non-unitary case, SR for the optimal choice of σ_n applied on the representation $\boldsymbol{D}^T\boldsymbol{y}$ approximates the MMSE pretty well.

In 4b we show the probability of recovering the true support P_{success} as a function of σ_n, both from (7) and from actual pursuit simulations. We also compare it to the MMSE weight from (2) and we marked the optimal MSE σ_n as a black line. We can see that the derived expression agrees with the simulations. In addition, we notice that the optimal σ_n in terms of MSE, also approximates the probability of the true support to the weight given in the MMSE solution. In other words, the optimal σ_n is the one that approximates the weight of the support to the weight given by the MMSE expression. The trend in (7) shows, unsurprisingly, that as we add noise, the probability of successfully recovering the true support decreases. In the limit, when $\sigma_n \to \infty$ the signal will be dominated by the noise and the success probability will be uniform among all the atoms, i.e. equal to $P_{success} = \frac{1}{m}$.



(a) 100 iterations of subtractive SR MSE with non unitary overcomplete dictionary and a single atom sparse representation.

(b) Numeric integration vs. simulations for probability $P(s = \text{True Suport}|y)$.

Fig. 4: One atom SR simulations with additive noise in the representation domain.

Due to the fact that the MSE is almost the same when the probability $P_{\text{SR}}(s = \text{True Support}|\boldsymbol{y}) \approx P_{\text{MMSE}}(s = \text{True Support}|\boldsymbol{y})$, one might expect a similar behavior for all the other possible supports. To emphasize this we draw the following experiment. We randomize an index and draw many signals $\boldsymbol{\alpha}$ with the non-zero in the same location. Then, we plot the histogram of the average empirical probability (obtained by pursuit) of each element in the vector $\boldsymbol{\alpha}$ to be non-zero. We compare these probabilities to those of the MMSE. This experiment will run for different σ_n values and each time we can compare the entire support histogram. We expect the two histograms (SR and MMSE) to fit for the right choice of

added noise σ_n. In Figure 5 we see the results of the described experiment.

Analyzing the results of this experiment, we notice that when no noise is added (this is the actually average case of the MAP estimator), most of the elements (apart from the true support element) have a much lower weight than the MMSE. As noise is added, the true support's probability decreases and its weight is divided among the other elements. At some point the two histograms almost match each other. This is the point where SR MSE almost equals that of the MMSE . As we add more noise, the true support's probability keeps decreasing and the other elements keep increasing and the histograms are now farther apart from each other. When we reach $\sigma_n \to \infty$ we obtain uniform probability for all the supports.

In Figure 6 we show on the left axis the $D_{K||L}$ distance (kullback-leibler divergence) between the two histograms, and on the right axis the MSE. We see, as expected, that when the histograms are close, the MSE is minimal.

We have shown empirically that SR approximates the MMSE well also when the dictionary is not unitary in the one atom case. We now need to find a way of estimating a proper σ_n. Using SURE is possible but seems impractical due to the complexity of the estimator.

2. Prior Assumptions on the Dictionary $\boldsymbol{D}$: In this section we will try to simplify the expression in (5) by adding assumptions regarding the dictionary $\boldsymbol{D}$. We will now show that if we assume that the columns of the dictionary, i.e. the atoms, are statistically uncorrelated, then adding noise in the image domain is the same as adding it to the representation. Formally, our assumption is that the atoms $\boldsymbol{d}_i$ are drawn from some random distribution that obeys the following properties:

$$\mathbb{E}\boldsymbol{d}_i^T\boldsymbol{d}_j = 0, \quad \forall i \neq j \quad 1 \leq i,j \leq m, \tag{8}$$

and of course that the atoms are normalized:

$$||\boldsymbol{d}_i||_2 = 1, \quad \forall i \quad 1 \leq i \leq m. \tag{9}$$

We now look to analyze the properties of the random vector in (5):

$$\tilde{\boldsymbol{n}} = \boldsymbol{D}^T(\boldsymbol{y}+\boldsymbol{n}).$$

First we observe that given the dictionary $\boldsymbol{D}$, each of the elements in this vector is asymptotically a Gaussian variable:

$$\tilde{n}_i|\boldsymbol{d}_i = \boldsymbol{d}_i^T(\boldsymbol{y}+\boldsymbol{n}) = \sum_{k=1}^{n} d_{i,k}(y_k+n_k) = \sum_{k=1}^{n} d_{i,k}y_k + \sum_{k=1}^{n} d_{i,k}n_k = \mu_i + \sum_{k=1}^{n} d_{i,k}n_k.$$

Given the measurements and the dictionary, the first sum $\sum_{k=1}^{n} d_{i,k}y_k \triangleq \mu_i$ is some constant. The second term in the expression is a weighted sum of n iid random variables $\{n_k\}_{k=1}^n$. Therefore, using the Central Limit Theorem, and the fact that $||d_i||_2 = 1$, asymptotically for large dimension n, this is a Gaussian variable. It is easy to see that its mean value is 0, and its standard deviation is σ_n, hence $\tilde{n}_i|\boldsymbol{d}_i \sim \mathcal{N}(\boldsymbol{d}_i^T\boldsymbol{y}, \sigma_n)$ for $n \to \infty$.

Now we turn to analyze the properties of the entire vector $\tilde{\boldsymbol{n}}$. From the previous analysis we know that given the dictionary $\boldsymbol{D}$, it is a random Gaussian vector with the mean vector $\boldsymbol{\mu}_{\tilde{\boldsymbol{n}}}|\boldsymbol{D} = \boldsymbol{D}^T\boldsymbol{y}$. Using the properties of the noise $\mathbb{E}\boldsymbol{n}\boldsymbol{n}^T = \sigma_n^2 I$, the auto-correlation matrix of $\tilde{\boldsymbol{n}}|\boldsymbol{D}$ is by definition:

$$\boldsymbol{\Sigma}|\boldsymbol{D} = \mathbb{E}\left[\boldsymbol{D}^T\boldsymbol{n}\boldsymbol{n}^T\boldsymbol{D}\middle|\boldsymbol{D}\right] = \boldsymbol{D}^T\mathbb{E}\left[\boldsymbol{n}\boldsymbol{n}^T\right]\boldsymbol{D} = \sigma_n^2\boldsymbol{D}^T\boldsymbol{D}.$$

Analyzing the average case, the mean vector is of the form:

$$\boldsymbol{\mu}_{\tilde{\boldsymbol{n}}} = \mathbb{E}_{\boldsymbol{D}}\boldsymbol{D}^T\boldsymbol{y},$$

and the auto-correlation matrix is simply diagonal:

$$\boldsymbol{\Sigma} = \mathbb{E}_{\boldsymbol{D}}[\boldsymbol{\Sigma}|\boldsymbol{D}] = \mathbb{E}\left[\sigma_n^2\boldsymbol{D}^T\boldsymbol{D}\right] = \sigma_n^2 I,$$

where we used the assumptions in (8) and (9). This means that the uncorrelated atoms assumption leads $\tilde{\boldsymbol{n}}$ to have the same properties as seen in the previous section, and therefore their analysis is the same.

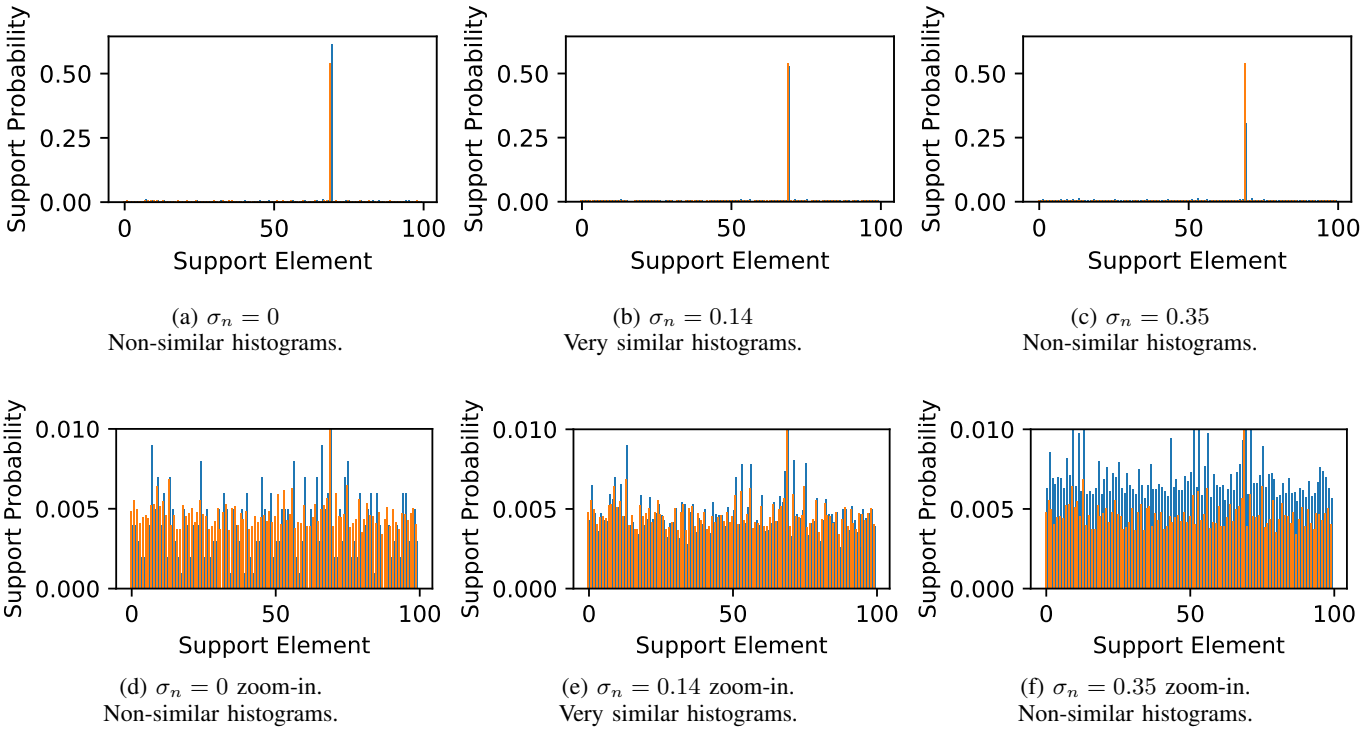
To show that practically the two are the same, we propose the following experiment. We sample a random dictionary $\boldsymbol{D}$ and random sparse representations $\boldsymbol{\alpha}$ with cardinality of 1 as the generative model described earlier suggests. In this experiment we used a dictionary $\boldsymbol{D}$ of size 200×400 and 2000 random sparse representations. Using the generated vectors and dictionary we created signals $\boldsymbol{y}$ simply by multiplying and adding noise $\boldsymbol{y} = \boldsymbol{D}\boldsymbol{\alpha} + \boldsymbol{\nu}$. To denoise the signals, we once run the stochastic resonance algorithm with noise $\boldsymbol{n}_1 \sim \mathcal{N}(0, \sigma_n I_{n\times n})$ added to the signal vectors $\boldsymbol{y}+\boldsymbol{n}_1$, and once with noise $\boldsymbol{n}_2 \sim \mathcal{N}(0, \sigma_n I_{m\times m})$ added to the representation domain $\boldsymbol{D}^T\boldsymbol{y}+\boldsymbol{n}_2$. Observe that due to the cardinality of the sparse representation, a simple Hard Thresholding is the MAP estimator, thus we shall use that as our non-linear estimator. In Figure 7 we see that the MSE of the two cases result in an almost identical curve. Small differences might exist due to the finite dimensions used in the experiment.

Note that the total noise energy added in the representation domain is much larger than that of the noise added to the signal, i.e. $E||n_2||_2^2 = m\sigma_n^2 > n\sigma_n^2 = E||n_1||_2^2$ but the results remain the same due to the unit norm of the dictionary $||d_i||_2 = 1$, and of course the uncorrelated atoms assumption.

To conclude this section, under the statistically uncorrelated atoms assumption, we have shown that adding noise in the signal domain $\boldsymbol{y}$, asymptotically converges to the analysis addressed in the previous section in which the noise was added to the representation domain $\boldsymbol{D}^T\boldsymbol{y}$. Therefore, under this assumption the previous section's analysis holds, and its results and conclusions can be inferred in this case as well.

B. Multiple Atoms

We shall now show that the application of Algorithm 1 in the general case of an overcomplete dictionary and many non-zeros results with superior denoising performance over the classic pursuit. We also show that this algorithm can be used with not only the l_0 pseudo-norm (OMP) but rather with l_1 norm algorithms (Basis Pursuit) as well. Unlike previous MMSE approximations under the sparseland model, such as the Random OMP, this is the first time an approximation using the l_1 norm is addressed. We use a random Gaussian dictionary and generate random Gaussian coefficients with



(a) $\sigma_n = 0$
Non-similar histograms.

(b) $\sigma_n = 0.14$
Very similar histograms.

(c) $\sigma_n = 0.35$
Non-similar histograms.

(d) $\sigma_n = 0$ zoom-in.
Non-similar histograms.

(e) $\sigma_n = 0.14$ zoom-in.
Very similar histograms.

(f) $\sigma_n = 0.35$ zoom-in.
Non-similar histograms.

Fig. 5: MMSE and 100 iterations of SR suppport weights histograms for varying values of σ_n;(a)-(c) show full histograms;(d)-(f) show zoomed-in histograms to emphasize the differences in the smaller weights. Atom number 69 is the true support.

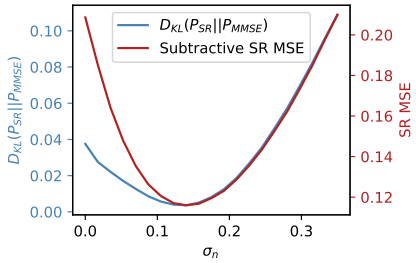


Fig. 6: Subtractive SR MSE and $D_{K||L}$ divergence between the MMSE and SR weights. When the divergence is small so is the MSE.

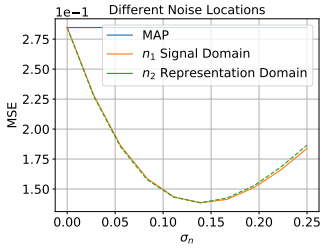


Fig. 7: Noise location comparison. 100 iterations of SR with OMP as a basic support estimator. $\boldsymbol{D} \in \mathbb{R}^{200\times 400}$ random dictionary. The measurements are $\boldsymbol{y} = \boldsymbol{D}\boldsymbol{\alpha} + \boldsymbol{\nu}$ where $\boldsymbol{\nu} \sim \mathcal{N}\left(\boldsymbol{0}, \sigma_{\boldsymbol{\nu}}^2 I\right)$, $\sigma_{\boldsymbol{\nu}} = 0.2$, $||\boldsymbol{\alpha}||_0 = 1$ and $\boldsymbol{\alpha}_s \sim \mathcal{N}(0,1)$. The SR noises are $\boldsymbol{n}_1 \sim \mathcal{N}(0, \sigma_n I_{n\times n})$. $\boldsymbol{n}_2 \sim \mathcal{N}(0, \sigma_n I_{m\times m})$.

a random number of non zeros and their locations, with an average sparsity of 5%. As in the previous experiments, we add Gaussian noise to the signal domain and use BP and OMP to denoise the signals. Since the number of non zeros is unknown, we use the bounded noise formulation of the pursuit algorithms, i.e.

$$\begin{aligned} &\text{(OMP)} \quad \min_{\boldsymbol{\alpha}} ||\boldsymbol{\alpha}||_0 \quad \text{s.t.} \quad ||\boldsymbol{y} - \boldsymbol{D}\boldsymbol{\alpha}||_2 \leq \epsilon, \\ &\text{(BP)} \quad \min_{\boldsymbol{\alpha}} ||\boldsymbol{\alpha}||_1 \quad \text{s.t.} \quad ||\boldsymbol{y} - \boldsymbol{D}\boldsymbol{\alpha}||_2 \leq \epsilon. \end{aligned}$$

The results of the described experiment can be seen in Figure 8. We acknowledge that although BP is in general inferior to OMP in terms of MSE, the SR method improves both algorithms' performance, indicating the indifferent of this concept to the pursuit algorithm used. We should mention that even though OMP achieved better MSE performance, as the number of non-zeros increases, BP converges much faster.

V. SR Extensions

In this section we represent some extensions to the basic concept introduced in Algorithm 1. The first is a link between SR and Monte Carlo Importance Sampling, and the second discusses whether Gaussian noise is the optimal choice.

A. SR With Monte Carlo Methods

In the case where the generative model is fully known, the MMSE can, in principle, be computed. The problem is that due to the huge amount of possible supports, it is impractical.

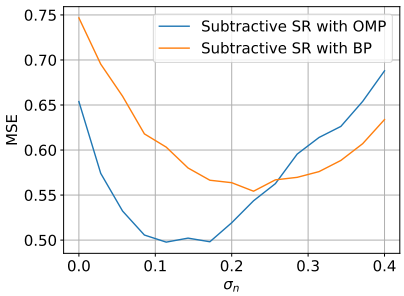


Fig. 8: Subtractive SR MSE with BP and OMP.

In order to solve this problem, we can turn to Monte Carlo simulations in order to achieve an approximation. As assumed previously in this work, the non-zeros come from a Gaussian distribution, and the additive noise is also Gaussian. Hence, the probabilities $P(S|\boldsymbol{y})$ have an exponential nature as shown in (2). This means that even though the true MMSE is a weighted sum of all possible supports:

$$\hat{\boldsymbol{\alpha}}(y) = \mathbb{E}[\boldsymbol{\alpha}|\boldsymbol{y}] = \mathbb{E}_{\mathbb{S}}\left[\mathbb{E}\left[\boldsymbol{\alpha}|\boldsymbol{y}, S\right]\right] = \sum_{S\in\Omega} P(S|y)\hat{\boldsymbol{\alpha}}(y,S),$$

the sum is practically dominated by only a few number of elements:

$$\hat{\boldsymbol{\alpha}}(y) = \sum_{S\in\Omega} P(S|y)\hat{\boldsymbol{\alpha}}(y,S) \approx \sum_{\substack{S\in\omega \\ \omega\subset\Omega}} P(S|y)\hat{\boldsymbol{\alpha}}(y,S),$$

for a proper choice of the subset Ω. Can we somehow find the significant elements, weight them accordingly and use them as an approximation for the MMSE?

1) Importance Sampling: As we have seen, the Bayesian approach requires a sum over all the possible supports (integration in the general Bayesian case) in order to achieve the MMSE estimator. In the literature, there are numerous ways of approximating non-analytic integrals [19]. Specifically for the posterior expectation case, the Monte Carlo Importance Sampling approach [20] is known to work well. Generally the integral we wish to approximate is:

$$\mathbb{E}_x\left[h(x)\right] = \int_{\mathcal{X}} h(x)f(x)dx.$$

Importance Sampling essentially calculates the above integral by using an additional sampling distribution g (also known as *importance sampling fundamental identity*):

$$\mathbb{E}_x\left[h(x)\right] = \int_{\mathcal{X}} h(x)f(x)dx = \int_{\mathcal{X}} h(x)\frac{f(x)}{g(x)}g(x)dx, \quad \{g(x)\neq 0 | x\in\mathcal{X}, f(x)\neq 0\}. \quad (10)$$

Note that the condition given in brackets means that the $\text{supp}(g) \supseteq \text{supp}(f)$. The above integral can be approximated by sampling J samples from the distribution g, i.e. $X_j \sim g, \quad 1\leq j\leq J$ and then average:

$$\mathbb{E}_x\left[h(x)\right] \approx \frac{1}{m}\sum_{X_j}\frac{f(x)}{g(x)}h(x). \quad (11)$$

This sum asymptotically converges to the integral in (10) by the Strong Law of Large Numbers. A well known alternative of the above sum is:

$$\mathbb{E}_x\left[h(x)\right] \approx \frac{\sum_{X_j}\frac{f(x)}{g(x)}h(x)}{\sum_{X_j}\frac{f(x)}{g(x)}}. \quad (12)$$

This formulation addresses some stability issues regarding the tail of f and g that are present in (11) and therefore is commonly used. As the previous sum, it also converges to (10) by the Strong Law of Large Numbers.

2) Importance Sampling using Stochastic Resonance: We propose to use stochastic resonance in order to retrieve the potentially important supports that have the dominant weights in the MMSE, and weight them accordingly using Importance Sampling. Formally, we use SR as a support generator PDF $S_j|\boldsymbol{y} \sim g(S|\boldsymbol{y})$, use the oracle estimators $h(\boldsymbol{\alpha}|S,\boldsymbol{y}) = \hat{\boldsymbol{\alpha}}^{\text{Oracle}}_{S_j,\boldsymbol{y}}$ and their MMSE un-normalized weights $f(S|\boldsymbol{y})$. Plugging this into (12) we get:

$$\hat{\boldsymbol{\alpha}}(\boldsymbol{y}) = \mathbb{E}\left[\boldsymbol{\alpha}|\boldsymbol{y}\right] \approx \frac{\sum_{S_j}\frac{f(S_j|\boldsymbol{y})}{g(S_j)}\hat{\boldsymbol{\alpha}}^{\text{Oracle}}_{S_j,\boldsymbol{y}}}{\sum_{S_j}\frac{f(S_j|\boldsymbol{y})}{g(S_j|\boldsymbol{y})}}, \quad S_j|\boldsymbol{y}\sim g(S|\boldsymbol{y}).$$

We now write the explicit expressions for each of the components described above:

- $\hat{\boldsymbol{\alpha}}^{\text{Oracle}}_{S,y}$ – is as stated in (1)
- $f(S|\boldsymbol{y})$ – This is the un-normalized probability for a support S given the noisy measurements vector $\boldsymbol{y}$. Again, given the described generative model and from (2)

$$P(S|\boldsymbol{y}) \propto f(S|\boldsymbol{y}) = t_S$$

- $g(S|\boldsymbol{y})$ – This is the support generator probability function given the noisy measurements $\boldsymbol{y}$. As previously mentioned we would like to have a way of generating probable supports. Using the SR concept we shall create likely supports simply by adding SR noise $\boldsymbol{n}$ to the measurements $\boldsymbol{y}$ and run a pursuit. Clearly, as we add more noise, in each of the iterations a pursuit might retrieve a different support. If we add too much noise then the supports recovered are not necessarily likely and we might miss the "preferred supports" with the highest MMSE weights. This means that another parameter for this generating function g is also the amount of noise to be added, σ_n.
  In order to quantify $g_{\sigma_n}(S|\boldsymbol{y})$ we simply use the empirical distribution. In other words, if we run K iterations and a specific support occurred k times, its probability is simply $g_{\sigma_n}(S|\boldsymbol{y}) = \frac{k}{K}$

By using the described components, we can now approximate the NP-Hard MMSE calculation simply by recovering the likely supports using any pursuit on the SR-noisy measurements and use (12). Note that this approximation will asymptotically converge to the MMSE with probability 1.

3) Results: Since asymptotically this method is guaranteed to converge to the MMSE, the question that arises is how fast do we converge versus the number of possible supports. We start by repeating the previously described experiments for comparison. Note that in this experiment there are only

100 possibilities for the support and therefore when running 100 iterations we are not surprised that we have successfully converged. These results can be seen in Figure 9a. To show the efficiency of this technique we compare it with the same settings, only this time the cardinality is $||\boldsymbol{\alpha}||_0 = 3$ giving it $\text{Tot}(S) = \binom{100}{3} = 161,700$ possibilities, all apriori equally likely! These results can be seen in Figure 9b. Note that the MAP and the MMSE are missing in this figure due to the impractical amount of calculations required. Compared to the previous SR methods described, this method is much less sensitive to the amount of added noise. As the noise increases we have minor degradation since the most likely supports are not recovered anymore. That being said, it seems that as the number of possible supports increases, it will take more iterations to actually converge or at least beat the previous SR method. In Figure 9c we see that after 200 iterations the two perform roughly the same, but Importance Sampling is much less sensitive to σ_n. Note that with 200 iterations we recover at most 200 different supports and that is only $\approx 0.12\%$ of all the possible supports.

To summarize, as we add more iterations we are guaranteed to asymptotically converge to the MMSE, but a major improvement can be achieved with a small amount of iterations. The biggest advantage of this method is its robustness to the energy of the SR noise.

B. What Noise Should be Used?

In the previous sections we used Gaussian noise by default. In this section we question this decision and wonder whether we can use noise models with different distributions and whether it affects the performance of the stochastic resonance estimator.

As mentioned briefly in IV-A1, the result of the additive noise multiplied by the dictionary $\boldsymbol{D}^T$ is Gaussian under mild conditions. Denoting $\tilde{\boldsymbol{n}} \triangleq \boldsymbol{D}^T\boldsymbol{n}$, each element $\tilde{n}_i$ is:

$$\tilde{n}_i = \boldsymbol{d}_i^T\boldsymbol{n} = \sum_{j=1}^{m} d_{i,j}n_j.$$

Without the loss of generality, assuming normalized atoms $||d_i||_2 = 1$, this expression a weighted average of m iid variables $\{n_j\}_{j=1}^m$. If the noise n_j has bounded mean and variance and, of course, assuming iid elements, we can use the Central Limit Theorem. In this case, for large enough signal dimensions, $\tilde{n}_i$ is asymptotically Gaussian regardless of the distribution of the original additive noise n.

Following the previous statement, we experiment with a different distribution for a random noise vector. We will employ an element-wise iid uniform noise with 0 mean $n_{\mathcal{U}} \sim \mathcal{U}[-r, r]$. In order to compare with a Gaussian noise $n_{\mathcal{N}} \sim \mathcal{N}(0, \sigma_n^2)$ we choose $r = \sqrt{3}\sigma_n$ thus assuring the same standard deviation for the two cases. Following the same experiment described in Figure 1a, in Figure 10a we also use a uniform noise distribution as described above. By using uniform noise with a zero mean and the same standard deviation as the Gaussian noise, we see a perfect match between the two curves. Due to the Central Limit Theorem, the noise's distribution will practically not change much as long as they uphold the described conditions, and the signal's dimensions are large enough.

Now that we know that the choice of the noise's distribution will not effect the performance, is there a distribution from which we can benefit more than others? To explore this question, consider the following. Given the signal $\boldsymbol{y}$, we define the subsampling noise $\boldsymbol{n}_{\text{subsample}}$ in the following way:

$$n_i(y_i) = \begin{cases} 0 & \text{w.p.} \quad p \\ -y_i & \text{w.p.} \quad 1-p \end{cases},$$

and the SR samples will now follow the following distribution:

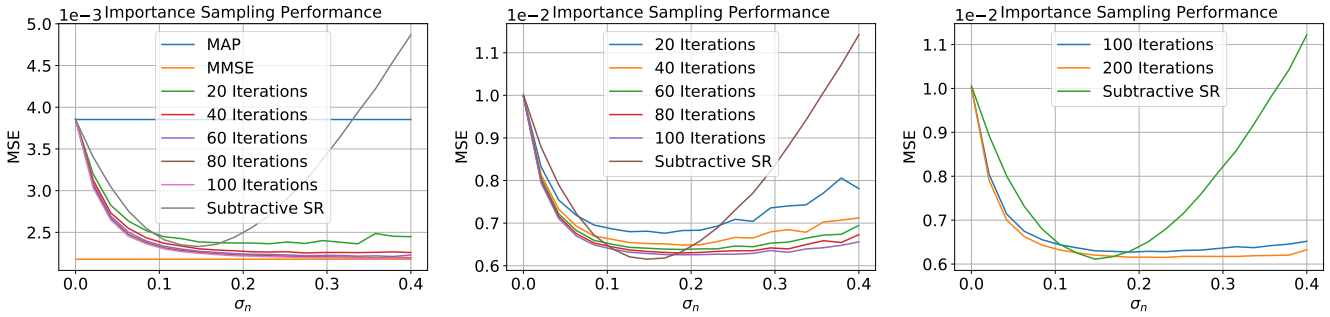
$$y_i + n_i(y_i) = \begin{cases} y_i & \text{w.p.} \quad p \\ 0 & \text{w.p.} \quad 1-p \end{cases}.$$

This means that each of the elements of the SR-noisy measurements will be zero with probability $1-p$ and only p measurements will remain in the signal. This distribution is interesting because of the following reason. When zeroing out an element in the vector $\boldsymbol{y}$, the matching row in in the dictionary $\boldsymbol{D}$ will always be multiplied by the zero element when calculating the correlations $\boldsymbol{D}^T\boldsymbol{y}_{\text{SR}}$ as done in most pursuits. This multiplication obviously has no contribution to the inner product and we might as well omit the zero elements from $\boldsymbol{y}_{\text{SR}}$ and the corresponding rows from $\boldsymbol{D}$, remaining with only a subsampled version of the signal $\boldsymbol{y}$ and the dictionary $\boldsymbol{D}$. In other words, in each of the SR iterations we simply subsample a random np portion from the signal $\boldsymbol{y}$ and the matching np portion of rows from the dictionary $\boldsymbol{D}$, remaining with $\boldsymbol{y}_{\text{subsample}}$ of size $pn \times 1$ and a dictionary $\boldsymbol{D}_{\text{subsample}}$ of size $pn \times m$ and apply a pursuit. Recall that once the pursuit is done, its result contains a noisy estimation of the signal due to the added SR noise $\boldsymbol{n}_{\text{subsample}}$. Just like in the previous cases we should now use the pursuit's result only as a support estimator in order to calculate the subtractive SR estimator. To do so, once the pursuit is done, we should turn back to the full sized signal $\boldsymbol{y}$ and dictionary $\boldsymbol{D}$ and calculate the oracle estimator using the support recovered by the subsampled pursuit.

The described process can be equally formulated by sampling a random mask $\boldsymbol{M}_{pn\times n}$ which is a random subsample of pn rows from the diagonal identity matrix $\boldsymbol{I}_{n\times n}$. In this formulation, we create many base estimators by applying a pursuit on the sub-sampled signal $\boldsymbol{y}_{\text{sub}} = \boldsymbol{M}\boldsymbol{y}$, using the sub-sampled dictionary $\boldsymbol{D}_{\text{sub}} = \boldsymbol{M}\boldsymbol{D}$. Note that when applying this method, each pursuit has a computational benefit over the previous methods due to the decreased size of the signal's dimension. In Figure 10b we show the results for the same experiment described in Figure 10a, this time using the subsampling approach. In this Figure the x axis represents the probability p, the percentage of elements kept in each pursuit. We see that the two methods have the same performance in MSE for the optimal choise of σ_n and p, but this method is faster due to the smaller size of the pursuit.

VI. Image Denoising

In this section we show the benefit of using SR for facial image denoising. In this experiment we use the Trainlets [21] dictionary trained on facial images from the Chinese Passport

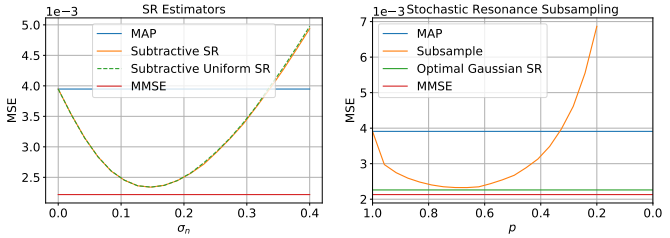


(a) 100 iterations of SR with OMP as a basic support estimator and {20,40,60,80,100} iterations of Importance Sampling approximation. $||\boldsymbol{\alpha}||_0 = 1$.

(b) 100 iterations of SR with OMP as a basic support estimator and {20,40,60,80,100} iterations of Importance Sampling approximation. $||\boldsymbol{\alpha}||_0 = 3$.

(c) 200 iterations of SR and {100,200} iterations of Importance Sampling approximation. $||\boldsymbol{\alpha}||_0 = 3$.

Fig. 9: SR with OMP as a basic support estimator and SR Importance Sampling approximation. $D_{50\times100}$ is an over-complete normalized non-unitary random dictionary. The measurements are $\boldsymbol{y} = \boldsymbol{D\alpha} + \boldsymbol{\nu}$ where $\boldsymbol{\nu} \sim \mathcal{N}\left(\mathbf{0}, \sigma_{\boldsymbol{\nu}}^2 I\right)$, $\sigma_{\boldsymbol{\nu}} = 0.2$ and $\boldsymbol{\alpha}_s \sim \mathcal{N}\left(0, \boldsymbol{I}\right)$.



(a) Uniform vs. Gaussian SR noise. $\boldsymbol{D} \in \mathbb{R}^{50\times100}$. Orange curve has Gaussian noise. Green curve has Uniform noise.

(b) Noise subsampling. $\boldsymbol{D} \in \mathbb{R}^{50\times100}$. Orange uses subsampling noise. Green shows optimal results with Gaussian noise.

Fig. 10: 100 iterations of SR with OMP as a basic support estimator. $\boldsymbol{D}$ is an over-complete normalized non-unitary random dictionary. The measurements are $\boldsymbol{y} = \boldsymbol{D\alpha} + \boldsymbol{\nu}$ where $\boldsymbol{\nu} \sim \mathcal{N}\left(\mathbf{0}, \sigma_{\boldsymbol{\nu}}^2 I\right)$, $\sigma_{\boldsymbol{\nu}} = 0.2$, $||\boldsymbol{\alpha}||_0 = 1$ and $\boldsymbol{\alpha}_s \sim \mathcal{N}\left(0, 1\right)$.

datase as described in [22]. In the dataset, each image is of size 100×100 pixels and contains a gray-scale aligned face. The application we demonstrate is denoising, that is approximating the following optimization problem:

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} ||\boldsymbol{D\alpha} - \boldsymbol{y}||_2 \quad \text{s.t.} \quad ||\boldsymbol{\alpha}||_0 = L.$$

The approximation is achieved using the Subspace Pursuit (SP) algorithm [23] which provides a fast converging algorithm for a fixed number of non-zeros L. The number of non-zeros L was achieved empirically and was chosen so that the denoising performance would be optimal. For Stochastic Resonance we used Algorithm 1 in its subtractive form with 200 iterations and the same SP settings. Note that we do not seek optimal denoising results but rather to show that SR can improve real image processing tasks.

A. Experiment Description

We corrupt an image from the dataset with Additive White Gaussian Noise (AWGN), each time with different standard deviation σ_ν. After that, for each image we applied SP using the Trainlets dictionary and for each σ_v choose L such that the denoised results would be optimal under the PSNR measure. After that we took the noisy images and used Algorithm 1 to denoise using the same L with varying SR noise standard deviations σ_n.

B. Results

In Figure 11 the results for $\sigma_\nu = 40$ can be seen. Unsurprisingly, the SR results has a clearer image with much less artifacts. Figure 12 presents the effectiveness of SR under varying SR noise σ_n. We see that a gain of almost 2 dB is achieved by using SR with a proper σ_n over the regular pursuit. Figure 13 presents a comparison of SP vs. SR for varying values of the noise's standard deviation σ_ν. In all of the described experiments, SR improved the denoising results. Generally we observe that as the noise is increased, the improvement is more significant.

VII. Conclusion

In this work we introduced Stochastic Resonance which is a phenomenon where noise improves the performance of a non-linear system. We suggested algorithms leveraging Stochastic Resonance under the context of sparse representation pursuit algorithms. We analyzed their theoretical properties under the SparseLand model setting and showed that MMSE approximation can be accomplished by repeatedly applying pursuits with different SR noise realizations, thus achieving many representations hypotheses and averaging all of them to a final dense estimator. We have demonstrated Stochastic Resonance as a practical and effective MMSE approximation that has the ability to use any pursuit algorithm as a "black box", thus opening the door for MMSE approximations in large dimensions.

(a) Noisy image. PSNR=16.1 dB. (b) Subspace Pursuit. PSNR=26.88 dB. (c) Stochastic Resonance. PSNR=28.76 dB. (d) Clean Image

Fig. 11: Denoising results comparison. $\sigma_\nu = 40$, $L = 90$.

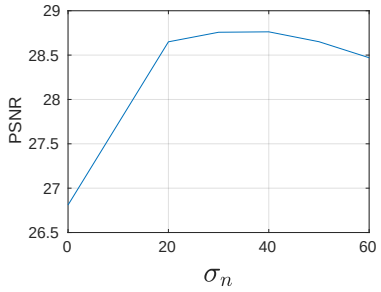


Fig. 12: SR results with varying σ_n for a noisy image with $\sigma_\nu = 40$, PSNR=16.1 dB. $\sigma_n = 0$ effectively does not use SR.

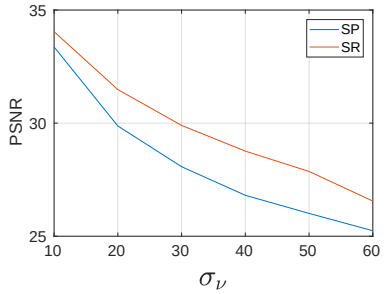


Fig. 13: SR and SP results comparison for varying standard deviation values σ_v.

Appendix A
Unitary SR Estimator Expectation

For the non-subtractive case:

$$\mathbb{E}_n\left[\mathcal{H}_\lambda\left(\beta+n\right)\right] = \int_{-\infty}^{\infty} \mathcal{H}_\lambda\left(\beta+n\right) p\left(n\right) dn$$

$$= \int_{|\beta+n|\geq\lambda} c^2\left(\beta+n\right) p\left(n\right) dn$$

$$= c^2\left[\int_{-\infty}^{-\lambda-\beta} \left(\beta+n\right) p\left(n\right) dn + \int_{\lambda-\beta}^{\infty} \left(\beta+n\right) p\left(n\right) dn\right]$$

$$= c^2\left[\int_{-\infty}^{-\lambda-\beta} \left(\beta+n\right) \frac{1}{\sqrt{2\pi\sigma_n^2}} e^{-\frac{n^2}{2\sigma_n^2}} dn\right] +$$

$$c^2\left[\int_{\lambda-\beta}^{\infty} \left(\beta+n\right) \frac{1}{\sqrt{2\pi\sigma_n^2}} e^{-\frac{n^2}{2\sigma_n^2}}\right]$$

$$= c^2\left[\beta Q\left(\frac{\lambda+\beta}{\sigma_n}\right) + \beta Q\left(\frac{\lambda-\beta}{\sigma_n}\right)\right] +$$

$$\left[\frac{\sigma_n}{\sqrt{2\pi}}\left(e^{-\frac{(\lambda-\beta)^2}{2\sigma_n^2}} - e^{-\frac{(\lambda+\beta)^2}{2\sigma_n^2}}\right)\right].$$

Similarly, for the subtractive case:

$$\mathbb{E}_n\left[\mathcal{H}^-\left(\beta,n\right)\right] = \int_{-\infty}^{\infty} \mathcal{H}^-\left(\beta+n\right) p\left(n\right) dn$$

$$= \int_{|\beta+n|\geq\lambda} c^2\beta p\left(n\right) dn$$

$$= c^2\left[\int_{-\infty}^{-\lambda-\beta} \beta p\left(n\right) dn + \int_{\lambda-\beta}^{\infty} \beta p\left(n\right) dn\right]$$

$$= c^2\beta\left[\int_{-\infty}^{-\lambda-\beta} \frac{1}{\sqrt{2\pi\sigma_n^2}} e^{-\frac{n^2}{2\sigma_n^2}} dn + \int_{\lambda-\beta}^{\infty} \frac{1}{\sqrt{2\pi\sigma_n^2}} e^{-\frac{n^2}{2\sigma_n^2}}\right]$$

$$= c^2\beta\left[Q\left(\frac{\lambda+\beta}{\sigma_n}\right) + Q\left(\frac{\lambda-\beta}{\sigma_n}\right)\right].$$

Appendix B
SURE Surface for the Unitary Case

Plugging in the subtractive estimator $\mathcal{H}^-$, into (3) results with the following expression:

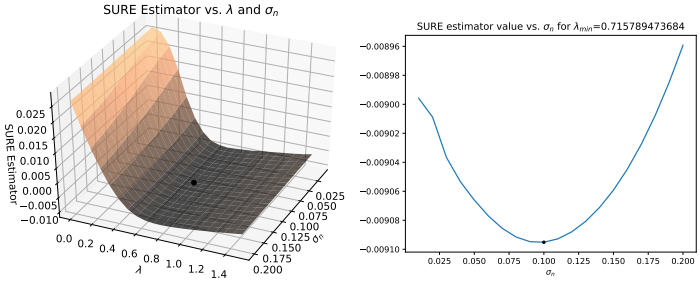
$$\mathbb{E}_n\mu\left(\mathcal{H}^-\right) = \sum_i \left(c^2\beta_i\left[Q\left(\frac{\lambda+\beta_i}{\sigma_n}\right) + Q\left(\frac{\lambda-\beta_i}{\sigma_n}\right)\right]\right)^2 -$$

$$\sum_i 2c^2\beta_i^2\left[Q\left(\frac{\lambda+\beta_i}{\sigma_n}\right) + Q\left(\frac{\lambda-\beta_i}{\sigma_n}\right)\right] +$$

$$\sum_i 2\sigma_\nu^2 c^2\left[Q\left(\frac{\lambda+\beta_i}{\sigma_n}\right) + Q\left(\frac{\lambda-\beta_i}{\sigma_n}\right)\right] +$$

$$\sum_i 2\sigma_\nu^2 c^2\beta_i\left[\frac{1}{\sqrt{2\pi}\sigma_n} e^{-\frac{(\lambda-\beta_i)^2}{2\sigma_n^2}} - \frac{1}{\sqrt{2\pi}\sigma_n} e^{-\frac{(\lambda+\beta_i)^2}{2\sigma_n^2}}\right]$$

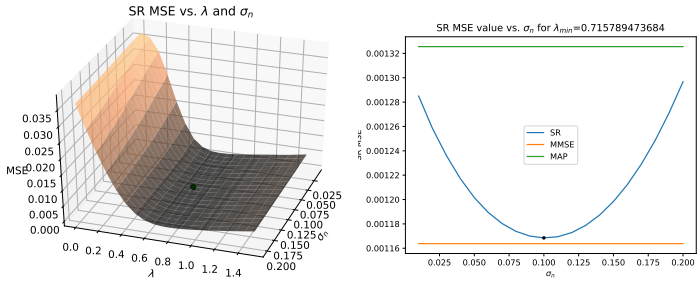
In order to show that it is indeed easy to optimize λ and σ on the SURE surface, we demonstrate it by the following experiment. We generated a sparse signal with probability of $P_i = 0.01$ for a non-zero. The non-zeros were generated randomly with a Gaussian distribution $\mathcal{N}\left(0,1\right)$. We projected the signal by a unitary dictionary and added random Gaussian noise $\mathcal{N}\left(0,0.2\right)$. Each signal has been estimated using the described subtractive estimator. Figure 14 shows the SURE surface over different λ and σ_n values, and Figure 15 shows the MSE results respectively. We can see that the SURE surface behaves just like the true MSE up to an additive constant and that it is smooth and rather easy to optimize. Also, in terms of MSE we see the obvious superiority of the proposed estimator over the MAP estimator, and that it is quite close to the MMSE.



(a) SURE Surface. The minimum is located at •

(b) SURE for a fixed optimal λ

Fig. 14: SURE values.



(a) MSE Surface. The minimum is located at •

(b) MSE for a fixed optimal λ. • represents the optimal value extracted using SURE.

Fig. 15: MSE values for λ and σ_n values.

Acknowledgment

The research leading to these results has received funding from the European Research Council under European Unions Seventh Framework Programme, ERC Grant agreement no. 320649.

References

[1] H. Chen, P. K. Varshney, and J. H. Michels, "Noise enhanced parameter estimation," *IEEE Transactions on Signal Processing*, vol. 56, no. 10 II, pp. 5074–5081, 2008.

[2] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on image processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[3] M. Elad and M. Aharon, "Image Denoising Via Sparse and Redundant Representations Over Learned Dictionaries," *IEEE TRANSACTIONS ON IMAGE PROCESSING*, vol. 15, no. 12, 2006.

[4] B. K. Natarajan, "SPARSE APPROXIMATE SOLUTIONS TO LINEAR SYSTEMS*," vol. 24, no. 2, pp. 227–234, 1995.

[5] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM review*, vol. 43, no. 1, pp. 129–159, 2001.

[6] J. A. Tropp and A. C. Gilbert, "Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit," *IEEE TRANSACTIONS ON INFORMATION THEORY*, vol. 53, no. 12, 2007.

[7] D. L. Donoho, M. Elad, and V. N. Temlyakov, "Stable recovery of sparse overcomplete representations in the presence of noise," *IEEE Transactions on information theory*, vol. 52, no. 1, pp. 6–18, 2006.

[8] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on information theory*, vol. 52, no. 2, pp. 489–509, 2006.

[9] J. A. Tropp, "Average-case analysis of greedy pursuit," in *Proc. of SPIE Vol*, vol. 5914, pp. 591412–1, 2005.

[10] J. S. Turek, I. Yavneh, and M. Elad, "On MMSE and MAP denoising under sparse representation modeling over a unitary dictionary," *IEEE Transactions on Signal Processing*, vol. 59, no. 8, pp. 3526–3535, 2011.

[11] M. Elad and I. Yavneh, "A Weighted Average of Sparse Representations is Better than the Sparsest One Alone," vol. 55, no. 10, pp. 1–35, 2009.

[12] P. Schniter, L. C. Potter, and J. Ziniel, "Fast bayesian matching pursuit," in *Information Theory and Applications Workshop, 2008*, pp. 326–333, IEEE, 2008.

[13] D. Middleton, I. of Electrical, and E. Engineers, *An introduction to statistical communication theory*. IEEE press Piscataway, NJ, 1996.

[14] W. Maass, "Noise as a resource for computation and learning in networks of spiking neurons," *Proceedings of the IEEE*, vol. 102, no. 5, pp. 860–880, 2014.

[15] F. Moss, L. M. Ward, and W. G. Sannita, "Stochastic resonance and sensory information processing: a tutorial and review of application," *Clinical neurophysiology*, vol. 115, no. 2, pp. 267–281, 2004.

[16] S. Kay, J. H. Michels, H. Chen, and P. K. Varshney, "Reducing Probability of Decision Error Using Stochastic Resonance," *IEEE SIGNAL PROCESSING LETTERS*, vol. 13, no. 11, 2006.

[17] H. Chen, L. R. Varshney, and P. K. Varshney, "Noise-enhanced information systems," *Proceedings of the IEEE*, vol. 102, no. 10, pp. 1607–1621, 2014.

[18] C. M. Stein, "Estimation of the mean of a multivariate normal distribution," *The annals of Statistics*, pp. 1135–1151, 1981.

[19] J. Stoer and R. Bulirsch, *Introduction to numerical analysis*, vol. 12. Springer Science & Business Media, 2013. Chapter 3: Topics in Integration, pp. 125-162.

[20] C. P. Robert and G. Casella., *Monte Carlo Statistical Methods*. The address: Springer, 2 ed., 2005. Chapter 3: Monte Carlo Integration, pp. 90–107.

[21] J. Sulam, B. Ophir, M. Zibulevsky, and M. Elad, "Trainlets: Dictionary learning in high dimensions," *IEEE Transactions on Signal Processing*, vol. 64, no. 12, pp. 3180–3193, 2016.

[22] J. Sulam and M. Elad, "Large inpainting of face images with trainlets," *IEEE signal processing letters*, vol. 23, no. 12, pp. 1839–1843, 2016.

[23] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE transactions on Information Theory*, vol. 55, no. 5, pp. 2230–2249, 2009.