# Normalized Iterative Hard Thresholding: Guaranteed Stability and Performance

Thomas Blumensath, *Member, IEEE*, and Mike E. Davies, *Member, IEEE*

*Abstract*—Sparse signal models are used in many signal processing applications. The task of estimating the sparsest coefficient vector in these models is a combinatorial problem and efficient, often suboptimal strategies have to be used. Fortunately, under certain conditions on the model, several algorithms could be shown to efficiently calculate near-optimal solutions. In this paper, we study one of these methods, the so-called Iterative Hard Thresholding algorithm. While this method has strong theoretical performance guarantees whenever certain theoretical properties hold, empirical studies show that the algorithm's performance degrades significantly, whenever the conditions fail. What is more, in this regime, the algorithm also often fails to converge. As we are here interested in the application of the method to real world problems, in which it is not known in general, whether the theoretical conditions are satisfied or not, we suggest a simple modification that guarantees the convergence of the method, even in this regime. With this modification, empirical evidence suggests that the algorithm is faster than many state-of-the-art approaches while showing similar performance. What is more, the modified algorithm retains theoretical performance guarantees similar to the original algorithm.

*Index Terms*—Sparse signal modeling, compressed sensing (CS), iterative hard thresholding, sparse inverse problems.

## I. INTRODUCTION

SPARSE signal models have been popular in signal processing for several years due to their applicability to a wide range of diverse signal processing problems, from source separation [2], denoising [3], coding [4], pattern recognition [5], to sampling [6].

The problem can be formulated as follows. Let $\mathbf{y} \in \mathbb{R}^M$ be an observed vector. Given a matrix $\mathbf{\Phi} \in \mathbb{R}^{M \times N}$, with more columns than rows, that is, with $N > M$, we are asked to find a vector $\mathbf{x}$ in which most elements are zero and such that $\mathbf{\Phi x}$ approximates $\mathbf{y}$, that is, such that $\|\mathbf{y} - \mathbf{\Phi x}\|_2$ is small.

Compressed sensing [6] is a special case of sparse signal modeling. In compressed sensing, we assume that a signal $f$ has an orthonormal expansion $f = \sum_{n=1}^{N} x_n \psi_n$, where most of the

$x_n$ are zero or negligibly small. Instead of sampling $f$ using traditional sampling theory using $N$ samples, compressed sensing proposes to use $M < N$ linear measurements $y_m = \langle \phi_m, f \rangle$. Collecting the observations into a vector, we can then write the sampling model as

$$\mathbf{y} = \mathbf{\Phi x} + \mathbf{e} \tag{1}$$

where $\mathbf{e}$ is observation noise and where $\mathbf{x}$ is the vector containing the elements $x_n$. The matrix $\mathbf{\Phi}$ has entries $\langle \phi_m, \psi_n \rangle$.

In order to recover the signal $f$ or, what is equivalent, the vector $\mathbf{x}$, we assume that $\mathbf{x}$ is sparse, that is, we have a sparse inverse problem. For example, under the assumption that we know that $\mathbf{x}$ is approximately $K$-sparse, that is, that $\mathbf{x}$ is well approximated by a vector with only $K$ nonzero elements, we could pose the following optimization problem:

$$\mathbf{x}^\star = \underset{\mathbf{x}\,:\,\|\mathbf{x}\|_0 \leq K}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{\Phi x}\|_2. \tag{2}$$

In words, we are looking for a vector that minimizes the approximation error $\|\mathbf{y} - \mathbf{\Phi x}\|_2$, but we constrain the search to those vectors $\mathbf{x}$, that have no more than $K$ nonzero elements. Note that the notation $\|\mathbf{x}\|_0$ refers to the number of nonzero elements in the vector $\mathbf{x}$.

Unfortunately, solving the above optimization is a combinatorial problem for which no universally efficient strategy is known. Instead, suboptimal algorithms are used. Importantly, under certain conditions (such as the restricted isometry property discussed below), many of these approaches are guaranteed to perform well. For example, convex optimization can be used to find near-optimal estimates of $\mathbf{x}$ [7] and similar bounds have been derived for estimates found by greedy algorithms such as Compressed Sensing Matching Pursuit (CoSaMP) [8], Subspace Pursuit [9], Iterative Hard Thresholding [10], and the method in [11]. One advantage of the greedy approaches is that they are often faster than the convex optimization methods. Furthermore, they can also be used to recover signals with more complex structures than sparsity, such as tree sparse signals [12].

The theoretical guarantees are predominantly worst case bounds and the condition under which the performance guarantees hold cannot be verified in a computationally efficient manner. Therefore, in practice, one typically runs the algorithms without explicit knowledge of whether the guarantees are satisfied or not. Luckily, many algorithms perform surprisingly well in a regime in which the theory does not hold and it is therefore imperative that the algorithms are robust to a violation of these theoretical conditions, that is, we at least require the algorithms to be stable.

While previous papers [13], [10] have derived strong theoretical guarantees for the Iterative Hard Thresholding algorithm, we here introduce a modification that guarantees stability, that is, that guarantees a decrease in the cost function in each iteration. We include a step size parameter and suggest an approach to determine the step size that guarantees that the normalized algorithm always converges. Importantly, with this modification, the algorithm is competitive to other state-of-the-art approaches. What is more, we also show that this modification still allows performance guarantees similar to those of the unmodified algorithm.

## II. ITERATIVE HARD THRESHOLDING

In a series of papers [14], [13], and [10], we studied theoretical properties of an algorithm termed Iterative Hard Thresholding ($\mathrm{IHT}_K$). This algorithm was inspired by the work in [15].

The algorithm proceeds as follows. Let $\mathbf{x}^0 = \mathbf{0}$ and use the iteration

$$\mathbf{x}^{n+1} = H_K(\mathbf{x}^n + \mathbf{\Phi}^T(\mathbf{y} - \mathbf{\Phi}\mathbf{x}^n)) \tag{3}$$

where $H_K(\mathbf{a})$ is the nonlinear operator that sets all but the largest (in magnitude) $K$ elements of $\mathbf{a}$ to zero. If there is no unique such set, a set can be selected either randomly or based on a predefined ordering of the elements.

### A. Conditions on $\mathbf{\Phi}$

All our results are based on properties of the matrix $\mathbf{\Phi}$ and in particular, on the so-called restricted isometry property (RIP) [6] of $\mathbf{\Phi}$. A matrix $\mathbf{\Phi}$ satisfies the (symmetric) RIP of order $K$, if there exists a $\delta_K < 1$ such that

$$(1 - \delta_K)\|\mathbf{x}\|_2^2 \leq \|\hat{\mathbf{\Phi}}\mathbf{x}\|_2^2 \leq (1 + \delta_K)\|\mathbf{x}\|_2^2 \tag{4}$$

for all vectors $\mathbf{x}$ with no more than $K$ nonzero elements. The smallest $\delta_K$ for which the above inequalities hold is called the *restricted isometry constant*.

As stated above, there is no computationally efficient algorithm to calculate the restricted isometry constant for a given matrix. However, it is known that, for various random constructions, matrices of certain dimensions have a small isometry constant with high probability. For example, if $M > cK \log(N/K)$ for some constant $c$, then, if the entries of $\mathbf{\Phi}$ are drawn from an independent and identically distributed (i.i.d.) Gaussian distribution with variance $1/M$, or if the entries of $\mathbf{\Phi}$ are either $1/\sqrt{M}$ or $-1/\sqrt{M}$ with equal probability, then the matrix will have a small restricted isometry constant with high probability [16], [17]. Exact expressions relating the constant $c$ and the probability with which $\delta_K$ attains a given value can be calculated explicitly.

### B. Convergence

Our previous papers concentrated on theoretical properties of the algorithm. In [14], convergence was studied and the following result obtained.

*Theorem 1:* If $\mathbf{\Phi}$ spans $\mathbb{R}^M$ and if $\|\mathbf{\Phi}\|_2 < 1$, then the Iterative Hard Thresholding algorithm converges to a local minimum

of the cost function $\|\mathbf{y} - \mathbf{\Phi}\mathbf{x}\|_2$ under the constraint that $\mathbf{x}$ is $K$-sparse.

This result guarantees stability of the algorithm in the sense that the approximation error is guaranteed to decrease and the estimate $\mathbf{x}^n$ converges to some fixed point. However, the theorem does not tell us anything about the quality of this fixed point.

### C. Signal Recovery Guarantee

A result that bounds the quality of the solution of the Iterative Hard Thresholding algorithm was derived in [10].

*Theorem 2:* Let $\mathbf{y} = \mathbf{\Phi}\mathbf{x} + \mathbf{e}$ be a given noisy observation, where $\mathbf{x}$ is an arbitrary vector. Let $\mathbf{x}^K$ be the best $K$-term approximation to $\mathbf{x}$. If $\mathbf{\Phi}$ has restricted isometry property with $\delta_{3K} < 1/\sqrt{32}$, then, at iteration $n$, $\mathrm{IHT}_K$ will recover an approximation $\mathbf{x}^n$ satisfying

$$\|\mathbf{x} - \mathbf{x}^n\|_2 \leq 2^{-n}\|\mathbf{x}^K\|_2 + 6\tilde{\epsilon}_K \tag{5}$$

where

$$\tilde{\epsilon}_K = \|\mathbf{x} - \mathbf{x}^K\|_2 + \frac{1}{\sqrt{K}}\|\mathbf{x} - \mathbf{x}^K\|_1 + \|\mathbf{e}\|_2. \tag{6}$$

Furthermore, after at most

$$n^\star = \left\lceil \log_2\left(\frac{\|\mathbf{x}^K\|_2}{\tilde{\epsilon}_K}\right) \right\rceil \tag{7}$$

iterations, $\mathrm{IHT}_K$ estimates $\mathbf{x}$ with accuracy

$$\|\mathbf{x} - \mathbf{x}^{n^\star}\|_2$$
$$\leq 7\left[\|\mathbf{x} - \mathbf{x}^K\|_2 + \frac{1}{\sqrt{K}}\|\mathbf{x} - \mathbf{x}^K\|_1 + \|\mathbf{e}\|_2\right]. \tag{8}$$

More generally, if $\delta_{3K} < 1/\sqrt{8}$, then there exists a finite constant $c$, such that

$$\|\mathbf{x} - \mathbf{x}^{n^\star}\|_2 \leq c\left[\|\mathbf{x} - \mathbf{x}^K\|_2 + \frac{1}{\sqrt{K}}\|\mathbf{x} - \mathbf{x}^K\|_1 + \|\mathbf{e}\|_2\right] \tag{9}$$

with $c \to \infty$ as $\delta_{3K} \to 1/\sqrt{8}$.

While this result does not guarantee that the algorithm converges to a single point, it nevertheless guarantees that the algorithm will reach a neighborhood of the best $K$-term approximation of any $\mathbf{x}$. How close the algorithm comes to this optimum depends on how well $\mathbf{x}$ can be approximated with a $K$-sparse vector.

### D. Discussion on These Results

Both, the symmetric restricted isometry property, as well as the Iterative Hard Thresholding algorithm are sensitive to a re-scaling of the matrix $\mathbf{\Phi}$. This is clearly an undesirable property. What is more, empirical evidence suggests that if the conditions in the above theorems fail, the $\mathrm{IHT}_K$ algorithm often becomes unstable.

To demonstrate this behavior, we conducted the following experiment. We generated 128 by 256 matrices $\mathbf{\Phi}$ (drawing the entries from i.i.d. Gaussian distributions) and generated $K$-sparse vectors by drawing the $K$ nonzero elements from an
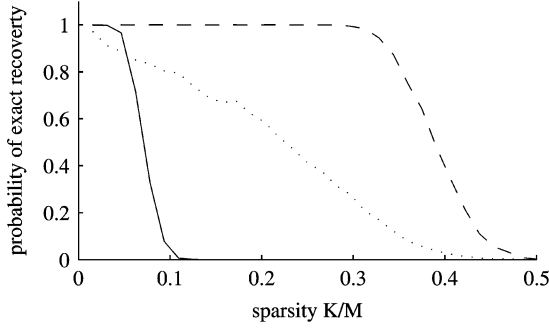
Fig. 1.  Exact recovery performance. Comparison between $\ell_1$ minimization (dashed), $\mathrm{IHT}_K$, where $\|\mathbf{\Phi}\|_2 = 1$ (dotted) and $\mathrm{IHT}_K$, where $\mathbf{\Phi}$ has unit norm columns and the algorithm is stopped when the cost $\|\mathbf{y} - \mathbf{\Phi x}\|_2$ increases (solid).

i.i.d. Gaussian distribution. Varying $K$ from 2 to 64 (in steps of 2) we averaged the results over 1000 problem realizations.

We then compared to different matrix normalizations, we normalized $\mathbf{\Phi}$ such that $\|\mathbf{\Phi}\|_2 = 1$ and we normalized the columns of $\mathbf{\Phi}$ so that they had unit length (which approximately ensures that the RIP condition is symmetric). The results of applying the Iterative Hard Thresholding algorithm are shown in Fig. 1, where we present the fraction of cases in which we could recover the true support of a $K$-sparse vector. The dotted line shows the results for the algorithm when $\|\mathbf{\Phi}\|_2 = 1$ and the solid line shows the results when $\mathbf{\Phi}$ had unit norm columns. When using the second type of normalization, we stopped the algorithm whenever the cost function in problem (2) increased to prevent instability. For comparison, we also show the results obtained by an $\ell_1$ minimization, that is, $\min_x \|\mathbf{x}\|_1 : \mathbf{y} = \mathbf{\Phi x}$, which is scaling-independent.

From these results it is clear that the performance of $\mathrm{IHT}_K$ varies significantly if different scalings of $\mathbf{\Phi}$ are used. Furthermore, for both scalings used here, the $\ell_1$ minimization results are significantly better. Note also that in this example, the theoretical condition on $\delta_{3K}$ that guarantees that both algorithms can recover the exact support breaks down for signals with as few as four nonzero elements so that the $\ell_1$-based method performs far beyond the region in which the theory holds. We therefore now derive a normalization of the $\mathrm{IHT}_K$ algorithm that not only makes the algorithms performance independent from arbitrary scaling of $\mathbf{\Phi}$, but will also make the algorithms empirical performance similar to that of the $\ell_1$ method.

## III. Normalized Iterative Hard Thresholding

In this section, we propose a simple modification of the algorithm[1] that guarantees convergence to a local minimum of the cost function, whatever the operator norm of $\mathbf{\Phi}$, while still retaining its performance guarantees if a restricted isometry property holds.

A closer look at the convergence proof in [14] reveals that the important step (and crucially, the step that relies on the singular values of $\mathbf{\Phi}$ to be strictly smaller than one) is that we need to

---

[1]This modification has now been incorporated into the latest version of the algorithm in the sparsify Matlab toolbox, which can be found on the first authors web page. The algorithm is accessible through the call to the function `hard_l0_Mterm`.

guarantee that $\|\mathbf{y} - \mathbf{\Phi x}^{n+1}\|_2^2 < (1-c)\|\mathbf{y} - \mathbf{\Phi x}^n\|_2^2$ in each iteration, for some $0 < c < 1$.

To motivate the following development, we recall the derivation of the Iterative Hard Thresholding algorithm in [14]. In order to optimize the cost function $\|\mathbf{y} - \mathbf{\Phi \hat{x}}\|_2^2$, under the constraint that $\|\hat{\mathbf{x}}\|_0 \le K$, we use a majorization minimization approach. The function $\|\mathbf{y} - \mathbf{\Phi \hat{x}}\|_2^2$ is majorized by

$$\|\mathbf{y} - \mathbf{\Phi \hat{x}}\|_2^2 + \|\hat{\mathbf{x}} - \mathbf{x}^n\|_2^2 - \|\mathbf{\Phi}(\hat{\mathbf{x}} - \mathbf{x}^n)\|_2^2 \qquad (10)$$

whenever $\|\mathbf{\Phi}\|_2 < 1$, that is, the minimizer $\mathbf{x}^{n+1}$ of (10) under the constraint that $\mathbf{x}$ is $K$-sparse satisfies $\|\mathbf{y} - \mathbf{\Phi x}^{n+1}\|_2^2 \le \|\mathbf{y} - \mathbf{\Phi x}^n\|_2^2$. To overcome the restriction on $\|\mathbf{\Phi}\|_2$, we use the following standard modification to the majorizing function:

$$\|\mu^{0.5}(\mathbf{y} - \mathbf{\Phi \hat{x}})\|_2^2 + \|\hat{\mathbf{x}} - \mathbf{x}^n\|_2^2 - \|\mu^{0.5}\mathbf{\Phi}(\hat{\mathbf{x}} - \mathbf{x}^n)\|_2^2. \quad (11)$$

For this to majorize $\|\mathbf{y} - \mathbf{\Phi \hat{x}}\|_2^2$ certain conditions on $\mathbf{\Phi}$ and $\mu$ must be satisfied, but, importantly, re-scaling of $\mathbf{\Phi}$ can be counteracted by an inverse scaling of $\mu$.

The above cost function is easy to minimize. Using arguments from [14], it can be seen that

$$\mathbf{x}^{n+1} = H_K(\mathbf{x}^n + \mu \mathbf{\Phi}^T(\mathbf{y} - \mathbf{\Phi x}^n)) \qquad (12)$$

calculates the minimum of this cost function under the constraint that $\mathbf{x}^{n+1}$ has to have no more than $K$ nonzero elements.

The original algorithm is a special case of this method where $\mu = 1$. Instead of constraining $\|\mathbf{\Phi}\|_2$ to guarantee convergence of the algorithm, we now suggest the following approach to determine $\mu$ adaptively, so that the algorithm becomes scale-independent.

Let $\Gamma^n$ be the support set of $\mathbf{x}^n$ and let $\mathbf{g} = \mathbf{\Phi}^T(\mathbf{y} - \mathbf{\Phi x}^n)$ be the negative gradient of $\|\mathbf{y} - \mathbf{\Phi x}\|_2^2$ evaluated at the current estimate $\mathbf{x}^n$. Should $\mathbf{x}^n$ be zero, which typically only happens in the first iteration when we initialize the algorithm with the zero vector, we use the index set of the largest $K$ (in magnitude) elements of $\mathbf{\Phi}^T \mathbf{y}$ as the set $\Gamma^n$. Assume that we had identified the correct support, that is, $\Gamma^n$ is the support of the best $K$ term approximation to $\mathbf{y}$. In this case, we would want to minimize $\|\mathbf{y} - \mathbf{\Phi}_{\Gamma^n} \mathbf{x}_{\Gamma^n}\|_2^2$. Using a gradient descent algorithm, this would be done using the iteration $\mathbf{x}_{\Gamma^n}^{n+1} = \mathbf{x}_{\Gamma^n}^n + \mu \mathbf{\Phi}_{\Gamma^n}^T(\mathbf{y} - \mathbf{\Phi}_{\Gamma^n} \mathbf{x}_{\Gamma^n}^n)$. Importantly, in the case in which the support is fixed, we can calculate an optimal step size, that is, a step size that maximally reduces the error in each iteration. It is easy to see that this step size is [18]

$$\mu = \frac{\mathbf{g}_{\Gamma^n}^T \mathbf{g}_{\Gamma^n}}{\mathbf{g}_{\Gamma^n}^T \mathbf{\Phi}_{\Gamma^n}^T \mathbf{\Phi}_{\Gamma^n} \mathbf{g}_{\Gamma^n}} \qquad (13)$$

where $\mathbf{g}_{\Gamma^n}$ is the sub-vector of $\mathbf{g}$ obtained by discarding all but the elements in $\Gamma^n$ and where $\mathbf{\Phi}_{\Gamma^n}$ is defined similarly by discarding columns of $\mathbf{\Phi}$. Evaluation of this quantity requires the calculation of $\mathbf{\Phi}_{\Gamma^n} \mathbf{g}_{\Gamma^n}$, the cost of which is equivalent to the evaluation of $\mathbf{\Phi x}^n$ and $\mathbf{\Phi}^T(\mathbf{y} - \mathbf{\Phi x}^n)$, so that the cost of the algorithm only increases by a constant fraction.

We here propose to use this step size also in the Iterative Hard Thresholding algorithm. In particular, in each iteration, we calculate $\mu$ as above and calculate a new proposition $\tilde{\mathbf{x}}^{n+1}$ using

(12). In the case in which the support of $\tilde{\mathbf{x}}^{n+1}$ is the same as that of $\mathbf{x}^n$, we are then guaranteed to have a maximal reduction in the cost function and we use $\mathbf{x}^{n+1} = \tilde{\mathbf{x}}^{n+1}$.

However, if the support of $\mathbf{x}^{n+1}$ differs from the support of $\mathbf{x}^n$, the optimality of $\mu$ is no longer guaranteed. In this case, a sufficient condition that guarantees convergence is (see Appendix A) that $\mu \leq \omega$, where

$$\omega = (1 - c)\frac{\|\tilde{\mathbf{x}}^{n+1} - \mathbf{x}^n\|_2^2}{\|\mathbf{\Phi}(\tilde{\mathbf{x}}^{n+1} - \mathbf{x}^n)\|_2^2} \qquad (14)$$

for a small fixed constant $c$.

Hence, if our first proposal $\tilde{\mathbf{x}}^{n+1}$ has a different support from the current estimate, we calculate $\omega$ and check whether $\mu < \omega$. If this holds, we keep the new update and set $\mathbf{x}^{n+1} = \tilde{\mathbf{x}}^{n+1}$. Otherwise we need to shrink the step size $\mu$. We here propose the update $\mu \leftarrow \mu/(\kappa(1-c))$, for some $\kappa > 1/(1-c)$. With this smaller step size, a new proposal $\tilde{\mathbf{x}}^{n+1}$ is calculated using (12). This procedure is terminated when $\mu < \omega$, in which case we accept the latest proposal and continue with the next iteration.

The algorithm is summarized as follows.

- Initialize $\mathbf{x}^1 = \mathbf{0}$, $\Gamma^1 = \operatorname{supp}(H_K(\mathbf{\Phi}^T \mathbf{y}))$,
- Iterate for $n = 1$, $i + +$, until stopping criterion is met
  - $\mathbf{g}^n = \mathbf{\Phi}^T(\mathbf{y} - \mathbf{\Phi}\mathbf{x}^n)$
  - $\mu^n = (\mathbf{g}_{\Gamma^n}^T \mathbf{g}_{\Gamma^n})/(\mathbf{g}_{\Gamma^n}^T \mathbf{\Phi}_{\Gamma^n}^T \mathbf{\Phi}_{\Gamma^n} \mathbf{g}_{\Gamma^n})$
  - $\tilde{\mathbf{x}}^{n+1} = H_K(\mathbf{x}^n + \mu^n \mathbf{g}^n)$.
  - $\Gamma^{n+1} = \operatorname{supp}(\tilde{\mathbf{x}}^{n+1})$
  - if $\Gamma^{n+1} = \Gamma^n$,
    - $\mathbf{x}^{n+1} = \tilde{\mathbf{x}}^{n+1}$,
  - else if $\Gamma^{n+1} \neq \Gamma^n$,
    - if $\mu^n \leq (1 - c)(\|\tilde{\mathbf{x}}^{n+1} - \mathbf{x}^n\|_2^2)/(\|\mathbf{\Phi}(\tilde{\mathbf{x}}^{n+1} - \mathbf{x}^n)\|_2^2)$
      - $\mathbf{x}^{n+1} = \tilde{\mathbf{x}}^{n+1}$,
    - else if $\mu^n > (1 - c)(\|\tilde{\mathbf{x}}^{n+1} - \mathbf{x}^n\|_2^2)/(\|\mathbf{\Phi}(\tilde{\mathbf{x}}^{n+1} - \mathbf{x}^n)\|_2^2)$,
      - iterate until $\mu^n \leq (1 - c)(\|\tilde{\mathbf{x}}^{n+1} - \mathbf{x}^n\|_2^2)/(\|\mathbf{\Phi}(\tilde{\mathbf{x}}^{n+1} - \mathbf{x}^n)\|_2^2)$,
        - $\mu^n \leftarrow \mu^n/(\kappa(1 - c))$
        - $\tilde{\mathbf{x}}^{n+1} = H_K(\mathbf{x}^n + \mu^n \mathbf{g}^n)$.
      - $\Gamma^{n+1} = \operatorname{supp}(\tilde{\mathbf{x}}^{n+1})$
      - $\mathbf{x}^{n+1} = \tilde{\mathbf{x}}^{n+1}]$,

### A. Bounds on $\mu$

An important question is whether the above algorithm is guaranteed to find a $\mu^n \leq \omega$. This will follow from a bound on $\mu$ which only depends on the following property of $\mathbf{\Phi}$. Let $\mathbf{\Phi}$ be such that $0 < \alpha_{2K} \leq \|\mathbf{\Phi}\mathbf{x}\|_2/\|\mathbf{x}\|_2 \leq \beta_{2K}$ holds for all vectors $\mathbf{x}$ with no more than $2K$ nonzero elements and let $\alpha_{2K}$ and $\beta_{2K}$ be the best possible constants in the above inequality. As $\mathbf{g}_{\Gamma^n}$ has only $K$ nonzero elements, we then have the trivial bound that

$$\frac{1}{\beta_{2K}^2} \leq \frac{1}{\beta_K^2} \leq \frac{\mathbf{g}_{\Gamma^n}^T \mathbf{g}_{\Gamma^n}}{\mathbf{g}_{\Gamma^n}^T \mathbf{\Phi}_{\Gamma^n}^T \mathbf{\Phi}_{\Gamma^n} \mathbf{g}_{\Gamma^n}} \leq \frac{1}{\alpha_K^2} \leq \frac{1}{\alpha_{2K}^2} \qquad (15)$$

which gives an upper bound on $\mu^n$.

We also know that $\omega \geq (1 - c)/(\beta_{2K}^2)$ so that any element $\mu \in ((1-c)/(\beta_{2K}^2), (1-c)/(\kappa(1-c)\beta_{2K}^2)]$ will satisfy $\mu \leq \omega$. If the algorithm selects $\mu^n = (\mathbf{g}_{\Gamma^n}^T \mathbf{g}_{\Gamma^n})/(\mathbf{g}_{\Gamma^n}^T \mathbf{\Phi}_{\Gamma^n}^T \mathbf{\Phi}_{\Gamma^n} \mathbf{g}_{\Gamma^n})$, then $\mu^n \geq (1)/(\beta_{2K}^2)$. Otherwise, the algorithm will select a $\mu^n \geq (1)/(\kappa\beta_{2K}^2)$. Using this lower bound we have

$$\frac{1}{\kappa\beta_{2K}^2} \leq \mu^n \leq \frac{1}{\alpha_{2K}^2} \qquad (16)$$

This implies that the algorithm is guaranteed to select a $\mu^n$ in no more than $\lfloor \log_{\kappa(1-c)}(\kappa\beta_{2K})/(\alpha_{2K}) \rfloor$ iterations.

Note that there is a tradeoff in the choice of $\kappa$. When taking smaller steps, we might have to take more steps in the line search, the theoretical results below will then however have somewhat better constants as the lower bound on $\mu^n$ will get closer to $(1)/(\beta_{2K}^2)$.

### B. Theoretic Properties of Normalized $\mathrm{IHT}_K$

The reason behind the condition that we require $\mu < \omega$ is that it guarantees that the cost $\|\mathbf{y} - \mathbf{\Phi}\mathbf{x}\|_2$ decreases from iteration to iteration and, what is more, as shown in Appendix A that it guarantees the convergence of the algorithm. Importantly, this holds irrespective of the value of the restricted isometry constant and the scaling of $\mathbf{\Phi}$. In particular, we prove the following theorem for our normalized algorithm.

*Theorem 3:* If $\operatorname{rank}(\mathbf{\Phi}) = M$ and $\operatorname{rank}(\mathbf{\Phi}_\Gamma) = K$ for all $\Gamma$ such that $|\Gamma| = K$, then the normalized Iterative Hard Thresholding algorithm converges to a local minimum of the optimization problem (2).

This convergence result guarantees stability of the algorithm for arbitrary scaling of $\mathbf{\Phi}$.

Importantly, we can also derive a result similar to Theorem 2 for the normalized algorithm, guaranteeing recovery performance in certain cases in which the restricted isometry property holds. As in [19], we here state this results in terms of a quantity $\gamma$ that is related to the restricted isometry property, but is scale-invariant.

*Theorem 3:* Assume $\mathbf{\Phi}$ satisfies the non-symmetric restricted isometry property

$$\alpha_{2K} \leq \frac{\|\mathbf{\Phi}\mathbf{x}\|_2}{\|\mathbf{x}\|_2} \leq \beta_{2K} \qquad (17)$$

for all $\mathbf{x} : \|\mathbf{x}\|_0 \leq 2K$. Given a noisy observation $\mathbf{y} = \mathbf{\Phi}\mathbf{x} + \mathbf{e}$, where $\mathbf{x}$ is an arbitrary vector, let $\mathbf{x}^K$ be the best $K$-term approximation to $\mathbf{x}$.

If the normalized Iterative Hard Thresholding algorithm has used $\mu = (\|\mathbf{g}_{\Gamma^n}\|_2^2)/(\|\mathbf{\Phi}_{\Gamma^n}\mathbf{g}_{\Gamma^n}\|_2^2)$ in each iteration, then let $\gamma_{2K} = (\beta_{2K}^2)/(\alpha_{2K}^2) - 1$ else let $\gamma_{2K} = \max\{1 - (\alpha_{2K}^2)/(\kappa\beta_{2K}^2), (\beta_{2K}^2)/(\alpha_{2K}^2) - 1\}$.

If $\gamma_{2K} < 1/8$, then, at iteration $n$, the approximation $\mathbf{x}^n$ satisfies

$$\|\mathbf{x} - \mathbf{x}^n\|_2 \leq 2^{-n}\|\mathbf{x}^K\|_2 + 8\tilde{\epsilon}_K \qquad (18)$$

where

$$\tilde{\epsilon}_K = \|\mathbf{x} - \mathbf{x}^K\|_2 + \frac{\|\mathbf{x} - \mathbf{x}^K\|_1}{\sqrt{K}} + \frac{1}{\beta_{2K}}\|\mathbf{e}\|_2. \qquad (19)$$

Furthermore, after at most $n^\star = \lceil \log_2(\|\mathbf{x}^K\|_2/\tilde{\epsilon}_K) \rceil$ iterations, $\text{IHT}_K$ estimates $\mathbf{x}$ with accuracy given by

$$\|\mathbf{x} - \mathbf{x}^{n^\star}\|_2 \leq 9\tilde{\epsilon}_K. \tag{20}$$

More generally, if $\gamma_{2K} < 1/4$, then there exists a finite constant $c$, such that

$$\|\mathbf{x} - \mathbf{x}^{n^\star}\|_2 \leq c\tilde{\epsilon}_K \tag{21}$$

with $c \to \infty$ as $\gamma_{2K} \to 1/4$.

The proof can be found in Appendix B.

This theorem is basically identical to Theorem 2, with the only difference being that we use the scale-invariant quantity $\gamma_{2K}$. The only part in the theorem that depends on the scaling of $\mathbf{\Phi}$ is the amplification of observation noise, which makes intuitive sense, because re-scaling $\mathbf{\Phi}$ means that we re-scale $\mathbf{\Phi x}$ so that the signal to noise ration $\|\mathbf{\Phi x}\|_2/\|\mathbf{e}\|_2$ increases or decreases accordingly.

### C. Monitoring the Algorithm

Letting $b^n = (\|\mathbf{\Phi}(\mathbf{x}^{n+1} - \mathbf{x}^n)\|_2^2)/(\|\mathbf{x}^{n+1} - \mathbf{x}^n\|_2^2)$, it is clear form the definition of the non-symmetric restricted isometry property that

$$\alpha_{2K}^2 \leq b^n \leq \beta_{2K}^2 \tag{22}$$

so that we can bound $\gamma_{2K}$ using

$$\gamma_{2K} \geq \max_{n,k} b^n/b^k - 1. \tag{23}$$

This offers a simple bound that can be used to monitor the algorithm. Because $\mathbf{\Phi x}^n$ has to be evaluated by the algorithm in each iteration, calculation of the bound can be done with negligible computational cost.

### D. Discussion on Step-Size Choice

We have above suggested a quite simple strategy to shrink $\mu^n$ to guarantee that $\mu^n \leq \omega$. An alternative approach would be the following. If the support of $\tilde{\mathbf{x}}^{n+1}$ differs from the support of $\mathbf{x}^n$ and if $\mu^n > \omega$, we could set $\mu^n = \omega$. This strategy has the following theoretical advantage. It is easily seen that this choice guarantees that

$$\mu^n \geq \frac{1-c}{\beta_{2K}} \tag{24}$$

which in turn implies that Theorem 4 holds with $\kappa = 1/(1-c)$, which would be the best choice possible and which obviously can't be achieved using $\mu^n \leftarrow \mu^n/(\kappa(1-c))$. On the other hand, setting $\mu^n = \omega$ does not guarantee that the algorithm will find a $\mu^n \leq \omega$ in a finite number of steps. A hybrid strategy might therefore be beneficial. However, we will not discuss such alternative strategy in any more detail and will use $\mu^n \leftarrow \mu^n/2$ in the experiments below.

### IV. SIMULATIONS

We are here particularly interested in the performance of the algorithm in a regime not covered by the theory. To this end,
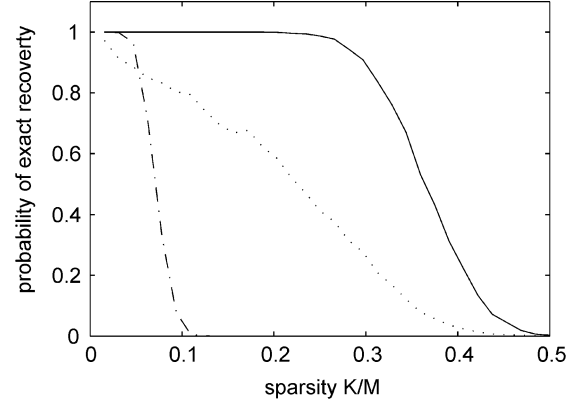


Fig. 2. Exact recovery performance. Comparison between the normalized IHT algorithm (solid), the unmodified algorithm where $\|\mathbf{\Phi}\|_2 = 1$ (dotted) and the unmodified algorithm, where $\mathbf{\Phi}$ has unit norm columns and the algorithm is stopped when the cost $\|\mathbf{y} - \mathbf{\Phi x}\|_2$ increases (dash-dotted).

we conducted several empirical studies. We first evaluate the approach on randomly generated artificial data before studying the performance on a larger problem from the Magnetic Resonance Imaging literature. The data in the next two subsections was generated as in Subsection II-D unless stated otherwise.

### A. Comparison to Unmodified Algorithm

We have demonstrated above that the original Iterative Hard Thresholding algorithm is sensitive to scaling of $\mathbf{\Phi}$. The new algorithm is now insensitive to such scaling and is guaranteed to be stable. We first compare the performance of the normalized algorithm to that of the original method. To do this, we repeated the experiment of Subsection II-D using the normalized algorithm. The results are shown in Fig. 2, where we show the percentage of cases in which the algorithms could recover the true support of a $K$-sparse vector. Here, the solid line is the performance of our normalized algorithm, while the other two lines are the results of the original algorithm with $\|\mathbf{\Phi}\|_2 = 1$ (dotted) and normalized columns (dash-dotted).

### B. Comparison to Other State-of-The-Art Methods

We here compare the empirical average performance of the normalized algorithm to the performance of two other state-of-the-art approaches, namely CoSaMP [8] and an $\ell_1$-based approach that minimizes $\|\mathbf{x}\|_1$ such that $\mathbf{y} = \mathbf{\Phi x}$. CoSaMP was stopped either when the estimate did not change significantly between iterations, or whenever the approximation error increased between iterations. This was required as CoSaMP is also not guaranteed to terminate if the restricted isometry property does not hold. We here used both an implementation of CoSaMP that used an exact least squares solution [8] (shown with dash dotted lines) and an implementation of CoSaMP in which we replace the exact least squares solution with several steps of gradient optimization as proposed in [8] (dotted lines, from left to right, using 3, 6, and 9 conjugate gradient iterations within each CoSaMP iteration). It is important to note that both versions of the CoSaAMP algorithm have the same theoretical guarantees as our Iterative Hard Thresholding algorithm [8].
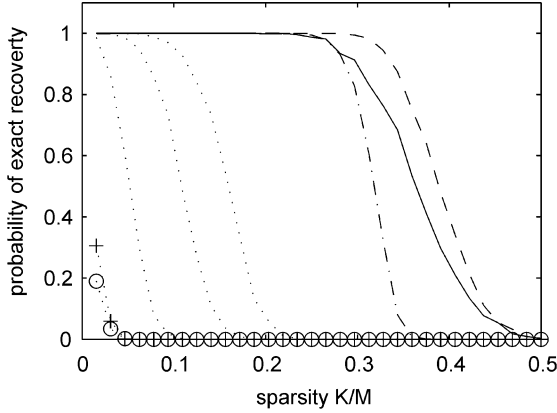
Fig. 3. Exact recovery performance where nonzero coefficients are Gaussian distributed. Comparison between normalized IHT (solid), $\ell_1$ solution (dashed), CoSaMP using the pseudo inverse (dash-dotted) and CoSaMP using (from left to right) 3, 6, or 9 conjugate gradient iterations (dotted). Also shown are the average cases in which the bound in Subsection III-C did not violate the requirement in the algorithm (dotted o) and the cases in which the algorithm used $\mu^{n+1} = (\|\mathbf{g}_\Gamma\|_2^2)/(\|\mathbf{\Phi}_\Gamma \mathbf{g}_\Gamma\|_2^2)$ in *all* iterations (dotted +).
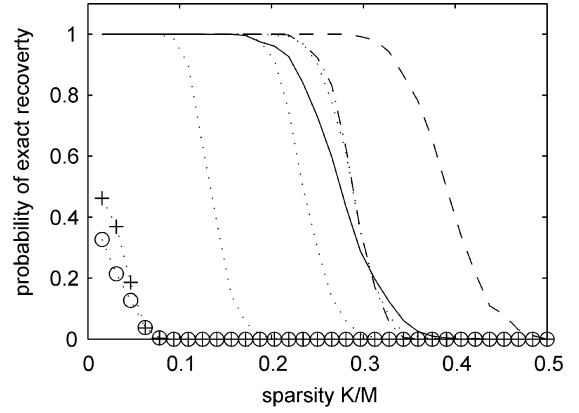


Fig. 4. Exact recovery performance where nonzero coefficients are 1. Comparison between normalized IHT (solid), $\ell_1$ solution (dashed), CoSaMP using the pseudo inverse (dash-dotted) and CoSaMP using (from left to right) 3, 6, or 9 conjugate gradient iterations (dotted). Also shown are the average cases in which the bound in Subsection III-C did not violate the requirement in the algorithm (dotted o) and the cases in which the algorithm used $\mu^{n+1} = (\|\mathbf{g}_\Gamma\|_2^2)/(\|\mathbf{\Phi}_\Gamma \mathbf{g}_\Gamma\|_2^2)$ in *all* iterations (dotted +).

The results are compared in Fig. 3, where we also show the number of cases for each $K$ for which we did not detect an RIP violation using the bound in Subsection III-C (dotted o) and the cases in which the algorithm used $\mu = (\|\mathbf{g}_\Gamma\|_2^2)/(\|\mathbf{\Phi}_\Gamma \mathbf{g}_\Gamma\|_2^2)$ in all iteration (dotted +).

Importantly, the algorithm performed well in this average case analysis far beyond the region where these two conditions are violated. In this example the normalized Iterative Hard Thresholding algorithm performed better than CoSaMP (with exact least squares) and nearly as well as the $\ell_1$-based approach. Also of interest is the observation that the efficient implementation of CoSaMP based on conjugate gradient updates performs significantly worse than the implementation based on the much slower exact least squares solution.

It is well known that many, though not all, algorithms for sparse signal recovery perform differently for different distributions of the nonzero coefficients. For example, many algorithms perform worse if the nonzero coefficients are all of equal magnitude. We therefore repeated the above experiment with the modification that we set the nonzero coefficients to 1. The recovery performance in this regime is shown in Fig. 4.

It is interesting to note the striking difference in the performance of CoSaMP when a few steps of a conjugate gradient solver were used. In this case, the performance of the algorithm for Gaussian coefficients was markedly worse than that observed for equal magnitude coefficients. This is in contrast to the performance of the normalized Iterative Hard Thresholding algorithm, which performed only modestly worse in the case of equal magnitude coefficients.

Fig. 5 shows the difference in computation time for the first experiment in the region where the algorithms perform well. It is important to realize that one of the implementations of CoSaMP (shown with dash dotted lines) requires the solution to an inverse problem in each iteration, which is costly in general. Normalized IHT on the other hand only requires the application of $\mathbf{\Phi}$ and $\mathbf{\Phi}^T$. In many practical situations, these matrices are designed to have fast implementations, based, for example, on the
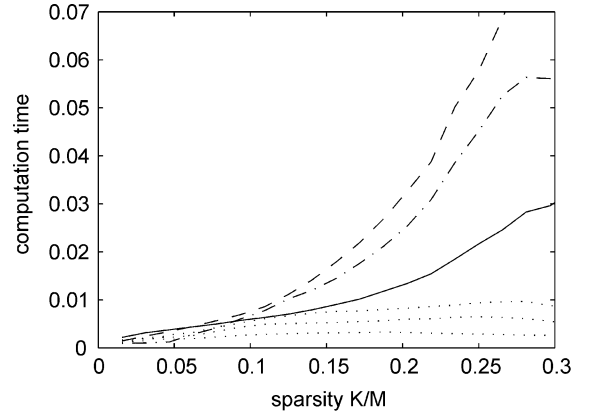


Fig. 5. Comparison between computation time for normalized IHT (solid), $\ell_1$ solution (dashed), CoSaMP using the pseudo inverse (dash-dotted) and CoSaMP using (from top to bottom) 9, 6, or 3 conjugate gradient iterations (dotted).

Fourier or wavelet transform. In this case, the computational benefit of the normalized IHT algorithm will be even greater and more dramatic than shown here for random matrices $\mathbf{\Phi}$. As noted above, if CoSaMP is implemented using conjugate gradient updates as proposed in [8], even though the algorithm's complexity decreases significantly, so does its performance. To solve the $\ell_1$ optimization problem, we here used the spgl1 [20] algorithm (available on (http://www.cs.ubc.ca/labs/scl/spgl1/).

### C. Influence of Noise

In the real world, signals are rarely exactly $K$-sparse. Furthermore, observations often have substantial noise contributions. The influence of these two effects on the performance of the algorithms compared in the previous subsection is studied here.

In the first experiment, we assumed that the observation was $\mathbf{y} = \mathbf{\Phi}\mathbf{x} + \mathbf{e}$, where $\mathbf{x}$ was $K$-sparse and where the noise term $\mathbf{e}$ was Gaussian. We normalized $\mathbf{\Phi}\mathbf{x}$ and $\mathbf{e}$ such that we had a specific signal to noise ratio (SNR). The performance for SNR values of 0, 10, 20, and 30 dB is shown in Fig. 6. We used the pseudo-inverse based implementation of CoSaMP. The $\ell_1$
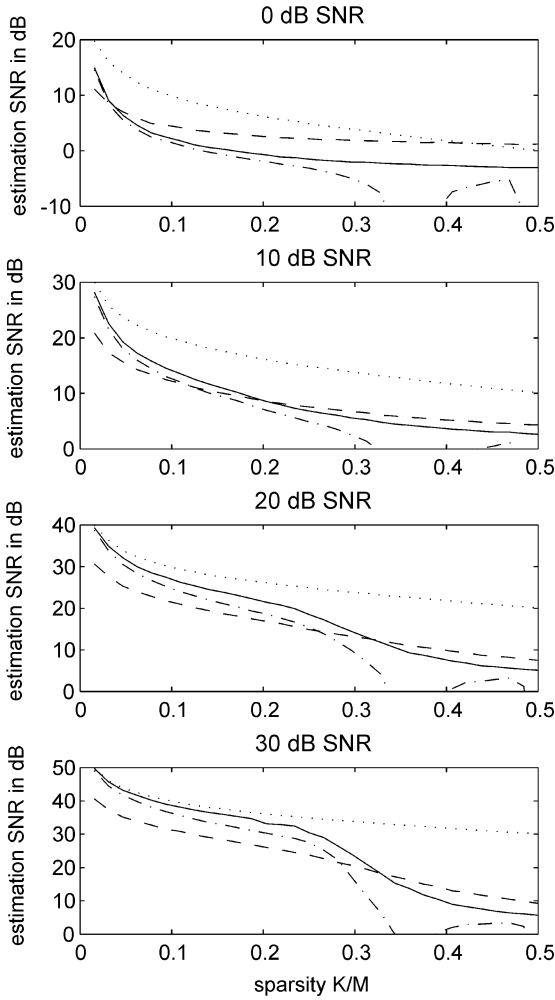
Fig. 6. Influence of observation noise. The observation was $\mathbf{y} = \mathbf{\Phi}\mathbf{x} + \mathbf{e}$, where $\mathbf{x}$ was $K$-sparse. We compare normalized IHT (solid), $\ell_1$ solution (dashed) and CoSaMP (dash-dotted). Also shown is an oracle estimate that knows the location of the nonzero coefficients (dotted).



Fig. 7. Compressible signals. Here $\mathbf{y} = \mathbf{\Phi}\mathbf{x}$, where $\mathbf{x}$ was generated using $x[n] = n^{-p}$, for $p \in \{1, 2, 4\}$. We compare normalized IHT (solid), $\ell_1$ solution (dashed) and CoSaMP (dash-dotted). Also shown is an oracle estimate that knows the location of the largest $K$ nonzero coefficients (dotted).

method used here solved $\min_{\hat{\mathbf{x}}} \|\hat{\mathbf{x}}\|_1$ under the constraint that $\|\mathbf{y} - \mathbf{\Phi}\hat{\mathbf{x}}\|_2 \leq \|\mathbf{e}\|_2$, where we assumed knowledge of $\|\mathbf{e}\|_2$. Also shown is the performance of an oracle estimator that knows the location of the $K$ largest coefficients in $\mathbf{x}$ and uses a least squares estimate for these coefficients (setting the remaining coefficients to zero).

The results show that, for moderate SNR values, such as 30 dB, the proposed algorithm is close to the optimal performance of the oracle and outperforms the other approaches, at least up to a sparsity of approximately 0.25. Interestingly, the $\ell_1$ solution, which we did not de-bias,[2] performed quite well for very low SNR values. Interesting here is that the $\ell_1$ solution even outperformed the oracle estimate in the 0 dB setting when $K/M$ was between 0.45 and 5. This is possibly due to the particular oracle estimate used. CoSaMP performed somewhat worse. Interestingly, once the signal became less sparse, CoSaMP did perform markedly worse than the other approaches, which is probably due to the fact that we had to

[2]That is, we did not calculate a least squares estimate, using the coefficients identified by the algorithm.
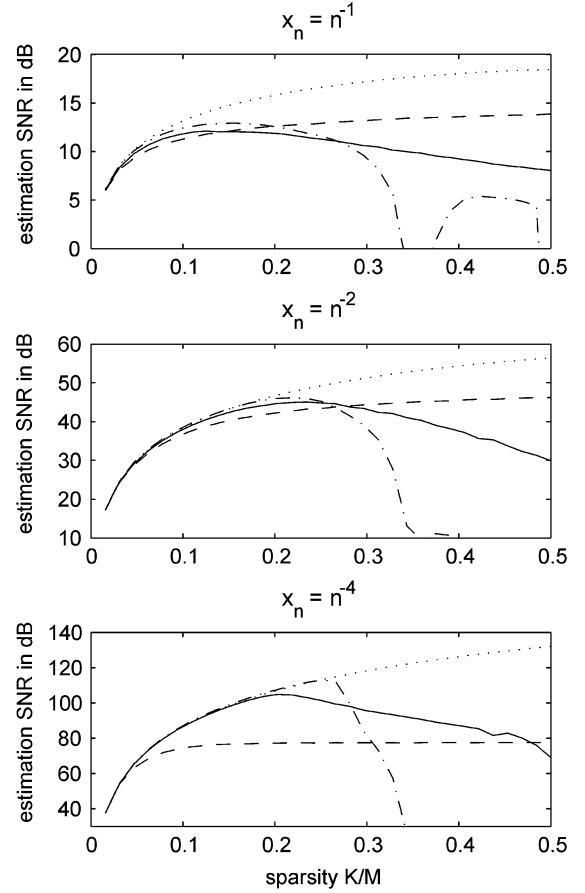
stop the algorithm whenever the approximation error increased to prevent instability.

In the next experiment (see Fig. 7), we studied the influence of noise in the signal domain, that is, $x$ was not exact sparse. We generated observations $\mathbf{y} = \mathbf{\Phi}\mathbf{x}$, where $\mathbf{x}$ was compressible, that is, the coefficients in $\mathbf{x}$ decayed with a power law. In particular, we generated $x[n] = n^{-p}$, where $p \in \{1, 2, 4\}$. In this experiment, we varied the sparsity of the solution for CoSaMP (which again used the pseudo-inverse) and normalized IHT. The $\ell_1$ method used here solved $\min_{\hat{\mathbf{x}}} \|\hat{\mathbf{x}}\|_1$ under the constraint that $\|\mathbf{y} - \mathbf{\Phi}\hat{\mathbf{x}}\|_2 < \|\mathbf{\Phi}(\mathbf{x}_K - \mathbf{x})\|_2$, where $\mathbf{x}$ is the vector used to generate the observation and $\mathbf{x}_K$ is the best $K$-term approximation to $\mathbf{x}$.

It is interesting to observe that the pseudo-inverse based CoSaMP remains close to the optimal performance for sparsities above those for which normalized IHT fails, however, once CoSaMP fails, the performance decreases rapidly with decreasing sparsity. Again, this is likely to be due to the fact that we had to stop CoSaMP whenever the approximation error increased to prevent instability.

### D. Application to Large Scale Problem

The next example demonstrates the applicability of the normalized algorithm to a somewhat larger problem. We take this
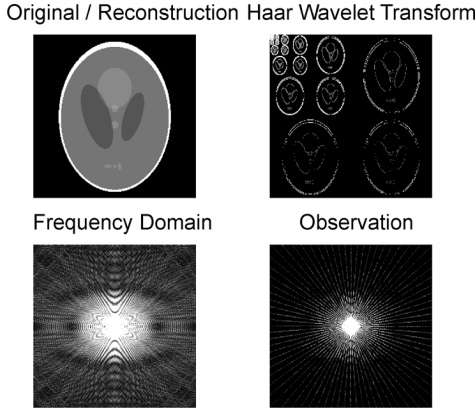
Original / Reconstruction Haar Wavelet Transform

Frequency Domain          Observation

Fig. 8. Magnetic Resonance Imaging example using the Shepp–Logan Phantom. Original and reconstructed Phantom (top left), Haar wavelet representation (top right), Fourier domain representation (bottom left), and radial slices through the Fourier domain (bottom right).



Fig. 9. Performance in recovering the Shepp–Logan phantom using radial line measurements. Ratio of measurements to signal dimension ($x$-axis) versus peak signal to noise ratio ($y$-axis) (thresholded at 130-dB PSNR). Results are shown for normalized IHT, Orthogonal Matching Pursuit (OMP), "Approximate" Conjugate Gradient Pursuit (GP) and $\ell_1$ optimization (L1).

example from the application domain of magnetic resonance imaging (MRI). In MRI, the scanner takes slices from the two-dimensional Fourier domain of the image [21]. In order to reduce scan time and the exposure of the patient to electromagnetic radiation, it is desirable to take fewer measurements. In this case, we can exploit the sparsity of the image in the wavelet domain and use compressed sensing methods to recover the image. In our notation, the observation vector $\mathbf{y}$ contains the measurements and the matrix $\boldsymbol{\Phi}$ models the inverse wavelet transform, the Fourier transform and the sub-sampling operator.

This is shown in Fig. 8, where the top right picture shows the wavelet transform of the image shown on the top left. This image is the standard Shepp–Logan phantom. The lower left picture shows the Fourier representation of the image and the lower right picture shows the measurements which here are radial slices through the Fourier domain.

We take between 30 and 52 radial slices from the Fourier domain and reconstruct the image from these measurements assuming sparsity in the Haar wavelet domain. The results are shown in Fig. 9, where we also show the results obtained with a range of other algorithms as reported previously in [22]. In particular, we show the results for the (approximate conjugate) Gradient Pursuit algorithm (GP) [22], the results obtained using Orthogonal Matching Pursuit (OMP) and an $\ell_1$-based approach minimising $\|\mathbf{y} - \boldsymbol{\Phi}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1$, where $\lambda$ was chosen experimentally so that the algorithm recovered approximately 4000 nonzero elements, which was the number of nonzero elements found in the wavelet representation of the original image. See [22] for more details on this experiment.

It can be seen that the Iterative Hard Thresholding algorithm performs as well as or better than the other approaches. It is also worth pointing out that we tried to run CoSaMP on this example. Due to the size of the problem, we had to use the implementation based on partial conjugate gradient solutions. However, we found that, if we stopped CoSaMP when the cost function increased, the results were very poor, while if we did not stop the algorithm whenever the cost function increased, the algorithm did not converge.
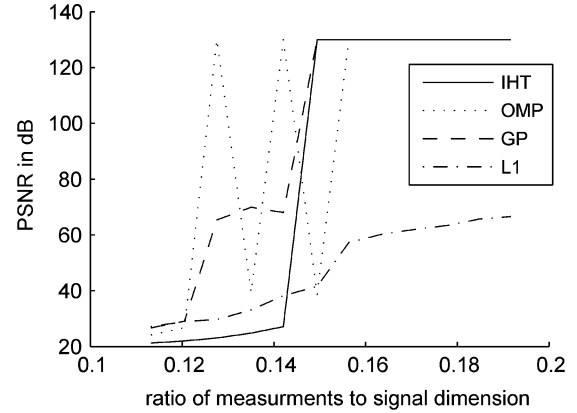
## V. CONCLUSION

The numerical studies of the Iterative Hard Thresholding algorithm reported in [14] were not very promising. This is due to the fact that in [14], the matrix $\boldsymbol{\Phi}$ was normalized such that its largest singular value was below 1 to guarantee stability. The theoretical analysis of the algorithm in [13] and [10] suggests that it is desirable that the algorithm satisfies the RIP condition, such that the singular values of all $3K$ element sub-matrices are *centred* around 1. A similar observation has been made here numerically. The problem however is that we then need to guarantee the stability of the algorithm. This can be done by the introduction of a simple adaptive step size and line search. With this modification, the algorithm is no longer dependent on the scaling of $\boldsymbol{\Phi}$ and its convergence is guaranteed. Furthermore, a recovery result can also be derived that guarantees that the result is near-optimal if $\boldsymbol{\Phi}$ satisfies a scale-independent restricted isometry property.

The normalized Iterative Hard Thresholding algorithm therefore offers a compromise that allows strong theoretical guarantees whenever $\boldsymbol{\Phi}$ satisfies a restricted isometry property with a small constant $\gamma_{2K}$. In the regime where this property fails, the algorithm is guaranteed to converge to a local minimum of the cost function (2) and empirically shows good average performance.

## APPENDIX A
## PROOF OF THEOREM 3

We start by proving the following lemma.

*Lemma 5:* Assume there is a subsequence of iterations $n_k$ for which the support of the estimate changes from one iteration to the next. Further assume that there exist a $\mu^{n_k+1}$ bounded from above and a constant $0 < c < 1$ such that the iteration

$$\mathbf{x}^{n_k+1} = H_K(\mathbf{x}_k^n + \mu^{n_k+1}\boldsymbol{\Phi}(\mathbf{y} - \boldsymbol{\Phi}\mathbf{x}^{n_k})) \qquad (25)$$

satisfies

$$\mu^{n_k+1} \leq (1-c) \frac{\|\mathbf{x}^{n_k+1} - \mathbf{x}^{n_k}\|_2^2}{\|\mathbf{\Phi}(\mathbf{x}^{n_k+1} - \mathbf{x}^{n_k})\|_2^2} \tag{26}$$

then, for each $\epsilon > 0$, there exist an integer $N$, such that for all $n_k > N$

$$\|\mathbf{x}^{n_k+1} - \mathbf{x}^{n_k}\|_2 \leq \epsilon. \tag{27}$$

*Proof:* Using the same argument as in [14], it is easy to show that the condition on $\mu^n$ guarantees that

$$\|\mathbf{y} - \mathbf{\Phi}\mathbf{x}^{n_k+1}\|_2 < \|\mathbf{y} - \mathbf{\Phi}\mathbf{x}^{n_k}\|_2 \tag{28}$$

and that

$$\|\mathbf{x}^{n_k+1} - \mathbf{x}^{n_k}\|_2^2 - \mu^{n+1}\|\mathbf{\Phi}(\mathbf{x}^{n_k+1} - \mathbf{x}^{n_k})\|_2^2$$
$$\leq \mu^{n_k+1}\|\mathbf{y} - \mathbf{\Phi}\mathbf{x}^{n_k}\|_2^2 - \mu^{n_k+1}\|\mathbf{y} - \mathbf{\Phi}\mathbf{x}^{n_k+1}\|_2^2. \tag{29}$$

Now consider the sequence $S_N = \sum_{k=0}^N \|\mathbf{x}^{n_k+1} - \mathbf{x}^{n_k}\|_2^2$, which is clearly increasing. Therefore, it converges if it is bounded, which will follow from the inequality $(1 - c)\|\mathbf{x}^{n_k+1} - \mathbf{x}^{n_k}\|_2^2 \geq \mu^{n_k+1}\|\mathbf{\Phi}(\mathbf{x}^{n_k+1} - \mathbf{x}^{n_k})\|_2^2$. In particular, we have

$$\sum_{k=0}^N \|\mathbf{x}^{n_k+1} - \mathbf{x}^{n_k}\|_2^2$$

$$\leq \frac{1}{c} \sum_{k=0}^N \left[ \|\mathbf{x}^{n_k+1} - \mathbf{x}^{n_k}\|_2^2 - \mu^{n_k+1}\|\mathbf{\Phi}(\mathbf{x}^{n_k+1} - \mathbf{x}^{n_k})\|_2^2 \right]$$

$$\leq \frac{1}{c} \sum_{k=0}^N \left[ \mu^{n_k+1}\|\mathbf{y} - \mathbf{\Phi}\mathbf{x}^{n_k}\|_2^2 - \mu^{n_k+1}\|\mathbf{y} - \mathbf{\Phi}\mathbf{x}^{n_k+1}\|_2^2 \right]$$

$$\leq \frac{1}{c_2} \sum_{k=0}^N \left[ \|\mathbf{y} - \mathbf{\Phi}\mathbf{x}^{n_k}\|_2^2 - \|\mathbf{y} - \mathbf{\Phi}\mathbf{x}^{n_k+1}\|_2^2 \right]$$

$$\leq \frac{1}{c_2} \|\mathbf{y} - \mathbf{\Phi}\mathbf{x}^{n_0}\|_2^2 - \|\mathbf{y} - \mathbf{\Phi}\mathbf{x}^N\|_2^2$$

$$\leq \frac{1}{c_2} \|\mathbf{y}\|_2^2.$$

The second inequality is (29) and the third inequality uses the fact that $\mu^{n_k+1}$ is bounded from above. The fourth inequality uses the fact that $\|\mathbf{y} - \mathbf{\Phi}\mathbf{x}^{n_k+1}\|_2 \leq \|\mathbf{y} - \mathbf{\Phi}\mathbf{x}^{n_{(k+1)}}\|_2$, so that we can cancel all but two terms in the summation.  ∎

Lemma 5 can now be used to prove convergence of the algorithm using the same argument as in [14].

There are two possibilities, either the support of $\mathbf{x}^n$ is the same for all $n > N$, for some $N$, or there exist infinitely many $n_k$ such that $\mathbf{x}^{n_k+1}$ and $\mathbf{x}^{n_k}$ have different support.

In the second case, we apply the following argument. By Lemma 5, we can choose any $\epsilon > 0$ and large enough $N$ such that for infinitely many $n_k > N$, $\|\mathbf{x}^{n_k+1} - \mathbf{x}^{n_k}\|_2 \leq \epsilon$ where $\mathbf{x}^{n_k+1}$ and $\mathbf{x}^{n_k}$ have different support. Looking at each element in these two vectors, we must have $|x_i^{n_k+1} - x_i^{n_k}| \leq \epsilon$, that is, the elements in $\mathbf{x}^{n_k}$ change by an arbitrarily small amount.

We now distinguish two cases.

Case 1: At least one zero element is set to a nonzero element. For this element, say $x_j$, because $x_j^{n_k} = 0$ and $x_j^{n_k+1} \neq 0$ the requirement that $|x_i^{n_k+1} - x_i^{n_k}| \leq \epsilon$ implies that $|x_j^{n_k+1}| \leq \epsilon$. By the definition of the operator $H_K$ used in the algorithm, this implies that all elements that are set to zero must satisfy

$$\left| x_i^{n_k} + \mu\phi_i^T(\mathbf{y} - \mathbf{\Phi}\mathbf{x}^{n_k}) \right| \leq \epsilon. \tag{30}$$

Furthermore, these elements must also satisfy

$$\left| x_i^{n_k+1} - x_i^{n_k} \right| = |x_i^{n_k}| \leq \epsilon. \tag{31}$$

This implies that for all elements that satisfy $|x_i^{n_k+1}| = 0$, we have

$$\left| \mu\phi_i^T(\mathbf{y} - \mathbf{\Phi}\mathbf{x}^{n_k}) \right| \leq |x_i^{n_k}| + \left| x_i^{n_k} + \mu\phi_i^T(\mathbf{y} - \mathbf{\Phi}\mathbf{x}^{n_k}) \right|$$
$$\leq 2\epsilon. \tag{32}$$

Similarly, for all elements that are nonzero in $\mathbf{x}^{n_k+1}$ we have $|x_i^{n_k+1} - x_i^{n_k}| = |\mu\phi_i^T(\mathbf{y} - \mathbf{\Phi}\mathbf{x}^{n_k})| \leq \epsilon$. Hence, in the case in which at least one zero element is set to a nonzero element, we are guaranteed that for all $i$

$$\left| \phi_i^T(\mathbf{y} - \mathbf{\Phi}\mathbf{x}^{n_k}) \right| \leq 2\epsilon/\mu. \tag{33}$$

Case 2: A nonzero element is set to 0 while no zero element is set to a nonzero element.

This can only happen if the size of the support set decreases. By the definition of the operator $H_K$ used in the algorithm, this implies that $x_i^{n_k} + \mu\phi_i^T(\mathbf{y} - \mathbf{\Phi}\mathbf{x}^{n_k}) = 0$ for all $i \in \Gamma_0$, where $\Gamma_0$ is the set of zero elements in $\mathbf{x}^{n_k}$. Therefore, for $i \in \Gamma_0$, we have

$$\left| x_i^{n_k+1} - x_i^{n_k} \right| = \left| \mu\phi_i^T(\mathbf{y} - \mathbf{\Phi}\mathbf{x}^{n_k}) \right| = 0. \tag{34}$$

In addition, for those $i$ for which $x_i^{n_k+1} = 0$ and $x_i^{n_k} \neq 0$, we have

$$x_i^{n_k} = -\mu\phi_i^T(\mathbf{y} - \mathbf{\Phi}\mathbf{x}^{n_k}) \tag{35}$$

and

$$\left| \mu\phi_i^T(\mathbf{y} - \mathbf{\Phi}\mathbf{x}^{n_k}) \right| = |x_i^{n_k}| = \left| x_i^{n_k+1} - x_i^{n_k} \right| \leq \epsilon. \tag{36}$$

Finally, for those $i$ for which both $x_i^{n_k+1} \neq 0$ and $x_i^{n_k} \neq 0$, we have

$$\left| x_i^{n_k+1} - x_i^{n_k} \right| = \left| \mu\phi_i^T(\mathbf{y} - \mathbf{\Phi}\mathbf{x}^{n_k}) \right| \leq \epsilon. \tag{37}$$

Therefore, we must have

$$\left| \phi_i^T(\mathbf{y} - \mathbf{\Phi}\mathbf{x}^{n_k}) \right| \leq \epsilon/\mu. \tag{38}$$

We have thus shown that for all $\epsilon > 0$ there is an $N$ such that for infinitely many $n_k > N$ for which $\mathbf{x}^{n_k+1}$ and $\mathbf{x}^{n_k}$ have different support

$$\left| \phi_i^T(\mathbf{y} - \mathbf{\Phi}\mathbf{x}^{n_k}) \right| \leq 2\epsilon/\mu$$

holds for all $i$.

We have shown that $\mu$ is bounded from below by a constant. Therefore, for these infinitely many $n_k$, letting $\epsilon \to 0$, we have $\|\mathbf{\Phi}^T(\mathbf{y} - \mathbf{\Phi}\mathbf{x}^{n_k})\|_2 \to 0$. As we assume that $\mathbf{\Phi}$ is full rank, we have some $\sigma > 0$ such that $\sigma\|(\mathbf{y} - \mathbf{\Phi}\mathbf{x}^{n_k})\|_2 \leq \|\mathbf{\Phi}^T(\mathbf{y} - \mathbf{\Phi}\mathbf{x}^{n_k})\|_2 \to 0$ which implies that the sequence $\{\|\mathbf{y} - \mathbf{\Phi}\mathbf{x}^{n_k}\|_2\}_k$ converges to zero.

We have therefore shown that $\{\|\mathbf{y} - \mathbf{\Phi}\mathbf{x}^n\|_2\}_n$ has a subsequence that converges to 0. Now $\|\mathbf{y} - \mathbf{\Phi}\mathbf{x}^n\|_2$ is decreasing and bounded, hence it also converges and as every subsequence of a convergent sequence converges to the same limit, it also converges to zero, that is, $\{\|\mathbf{y} - \mathbf{\Phi}\mathbf{x}^n\|_2\}_n \to \mathbf{0}$. Note that this is only possible if $\mathbf{x}^n$ (which is a $K$-sparse vector) converges to a $K$-sparse vector $\mathbf{x}$ for which $\mathbf{\Phi}\mathbf{x} = \mathbf{y}$.

Hence, if the support changes infinitely often, then the algorithm converges to a $K$-sparse vector for which $\mathbf{\Phi}\mathbf{x} = \mathbf{y}$. This is clearly a global minimum of the cost-function $\|\mathbf{y} - \mathbf{\Phi}\mathbf{x}\|_2$ which satisfies the constraint that $\mathbf{x}$ is $K$-sparse.

It therefore remains to analyse what happens if the support of $\mathbf{x}^n$ remains fixed for $n > N$ for some $N$.

In this case our algorithm reduces to a simple gradient descend algorithm that minimizes $\mathbf{y} - \mathbf{\Phi}_\Gamma \mathbf{x}_\Gamma$, where $\Gamma$ is the support set which remains fixed after some iteration. We further assume that $\mathbf{\Phi}_\Gamma$ is of rank $K$. In this case, convergence is well known. See for example [18, p. 521, 3rd. ed.], which shows the linear convergence of the algorithm to $\mathbf{x}^\star = (\mathbf{\Phi}_\Gamma^T\mathbf{\Phi}_\Gamma)^{-1}\mathbf{\Phi}_\Gamma^T\mathbf{y}$.

It remains to show that $\mathbf{x}^\star$ is a local minimum of the cost function $\|\mathbf{y} - \mathbf{\Phi}\mathbf{x}\|_2$ under the constraint that $\mathbf{x}$ has to be $K$-sparse. This has already be established in [14] where we have shown the following.

*Lemma 6:* A fixed point $\mathbf{x}^\star$ is a local minimum of the cost function $\|\mathbf{y} - \mathbf{\Phi}\mathbf{x}\|_2$ under the constraint that $\mathbf{x}$ is $K$-sparse.

## APPENDIX B
## SIGNAL RECOVERY PERFORMANCE

Assume $\mathbf{\Phi}$ satisfies the restricted isometry property

$$0 < \alpha_K \leq \frac{\|\mathbf{\Phi}\mathbf{x}\|_2}{\|\mathbf{x}\|_2} \leq \beta_K \tag{39}$$

for all $\mathbf{x} : \|\mathbf{x}\|_0 \leq K$.

If the normalized Iterative Hard Thresholding algorithm has used $\mu = (\|\mathbf{g}_{\Gamma^n}\|_2^2)/(\|\mathbf{\Phi}_{\Gamma^n}\mathbf{g}_{\Gamma^n}\|_2^2)$ in each iteration, then $(1)/(\beta_{2K}^2) \leq \mu^n \leq (1)/(\alpha_{2K}^2)$, else we have using the lower bound on $\mu^n$ that $(1)/(\kappa\beta_{2K}^2) \leq \mu^n \leq (1)/(\alpha_{2K}^2)$. Using these bounds on $\mu^n$ together with the arguments in [8] the following lemma can be derived

*Lemma 7:* If $\mathbf{\Phi}$ satisfies the restricted isometry with constants $\alpha_{2K}$ and $\beta_{2K}$ and let $\gamma_{2K}$ be as in the theorem, then for $|\Gamma| \leq 2K$

$$\|\mathbf{\Phi}_\Gamma^T\mathbf{y}\|_2 \leq \beta_{2K}\|\mathbf{x}\|_2 \tag{40}$$

$$\|\mu^n\mathbf{\Phi}_\Gamma^T\mathbf{\Phi}_\Gamma\mathbf{x}_\Gamma\|_2 \leq (\gamma_{2K} + 1)\|\mathbf{x}_\Gamma\|_2 \tag{41}$$

and

$$\left\|(\mathbf{I} - \mu^n\mathbf{\Phi}_\Gamma^T\mathbf{\Phi}_\Gamma)\mathbf{x}_\Gamma\right\|_2 \leq \gamma_{2K}\|\mathbf{x}_\Gamma\|_2. \tag{42}$$

Also, for two *disjoint* sets $\Gamma$ and $\Lambda$ (i.e., $\Gamma\bigcap\Lambda = \emptyset$) with $|\Gamma\bigcup\Lambda| \leq 2K$

$$\left\|\mu^n\mathbf{\Phi}_\Gamma^T\mathbf{\Phi}_\Lambda\mathbf{x}_\Lambda\right\|_2 \leq \gamma_{2K}\|\mathbf{x}_\Lambda\|_2. \tag{43}$$

We also use a result similar to [8, Prop. 3.5]

*Lemma 8:* For any $\mathbf{x}$, let $\mathbf{x}^K$ be the best $K$-term approximation to $\mathbf{x}$. Let $\Gamma$ be any $K$ element subset.

If the RIP holds for sparsity $K$, then

$$\left\|\mu^n\mathbf{\Phi}_\Gamma^T\mathbf{\Phi}(\mathbf{x} - \mathbf{x}^K)\right\|_2$$
$$\leq (\gamma_{2K} + 1)\left[\|\mathbf{x} - \mathbf{x}^K\|_2 + \frac{\|\mathbf{x} - \mathbf{x}^K\|_1}{\sqrt{K}}\right] \tag{44}$$

*Proof:* Let $T_0$ be the set of the $K$ largest (in magnitude) elements in $\mathbf{x}$, let $T_1$ be the set of the next $K$ largest elements and so on and let $\Gamma$ be any $K$ element subset. We have [23]

$$\left\|\mu^n\mathbf{\Phi}_\Gamma^T\mathbf{\Phi}(\mathbf{x} - \mathbf{x}^K)\right\|_2$$
$$\leq \sum_{i=1}^L \|\mu^n\mathbf{\Phi}_\Gamma^T\mathbf{\Phi}_{T_i}\mathbf{x}_{T_i}\|_2$$
$$\leq (\gamma_{2K} + 1)\sum_{i=1}^L \|\mathbf{x}_{T_i}\|_2$$
$$\leq (\gamma_{2K} + 1)\left[\|\mathbf{x}_{T_1}\|_2 + \sum_{i=2}^L \sqrt{K}\|\mathbf{x}_{T_i}\|_\infty\right]$$
$$\leq (\gamma_{2K} + 1)\left[\|\mathbf{x}_{T_1}\|_2 + \frac{1}{\sqrt{K}}\sum_{i=2}^L \|\mathbf{x}_{T_{i-1}}\|_1\right]$$
$$\leq (\gamma_{2K} + 1)\left[\|\mathbf{x}_{T_1}\|_2 + \frac{\sum_{i=1}^L \|\mathbf{x}_{T_i}\|_1}{\sqrt{K}}\right]. \tag{45}$$

$\blacksquare$

We use the following notation:
1) $\mathbf{y} = \mathbf{\Phi}\mathbf{x} + \mathbf{e} = \mathbf{\Phi}\mathbf{x}^K + \mathbf{\Phi}(\mathbf{x} - \mathbf{x}^K) + \mathbf{e} = \mathbf{\Phi}\mathbf{x}^K + \tilde{\mathbf{e}}$;
2) $\Gamma^n = \text{supp}\{\mathbf{x}^n\}$;
3) $B_{12} = \text{supp}\{\mathbf{x}^{n+1} - \mathbf{x}^K\}$;
4) $B_3 = \text{supp}\{\mathbf{x}^n - \mathbf{x}^K\} \setminus B_{12}$ (note that $|B_3| \leq K$);
5) $\mathbf{r}^n = \mathbf{x}^K - \mathbf{x}^n$;
6) $\mathbf{g} = \mathbf{\Phi}^T(\mathbf{y} - \mathbf{\Phi}\mathbf{x}^n)$;
7) $\mathbf{a}^{n+1} = \mathbf{x}^n + \mu^n\mathbf{g}$.

Using the triangle inequality, we first bound

$$\|\mathbf{x} - \mathbf{x}^n\|_2 \leq \|\mathbf{x}^K - \mathbf{x}^n\|_2 + \|\mathbf{x} - \mathbf{x}^K\|_2 \tag{46}$$

and then bound

$$\|\mathbf{x}^K - \mathbf{x}^{n+1}\|_2 \leq \left\|\mathbf{x}_{B_{12}}^K - \mathbf{a}_{B_{12}}^{n+1}\right\|_2 + \left\|\mathbf{x}_{B_{12}}^{n+1} - \mathbf{a}_{B_{12}}^{n+1}\right\|_2.$$

We know that $\mathbf{x}^{n+1}$ is a better $K$-term approximation to $\mathbf{a}_{B_{12}}^{n+1}$ than $\mathbf{x}^K$ so that

$$\|\mathbf{x}^K - \mathbf{x}^{n+1}\|_2 \leq 2\|\mathbf{x}_{B_{12}}^K - \mathbf{a}_{B_{12}}^{n+1}\|_2. \tag{47}$$

Expanding $\mathbf{a}^{n+1}$ we have

$$
\begin{aligned}
\|\mathbf{x}^K &- \mathbf{x}^{n+1}\|_2 \\
&\leq 2 \left\| \mathbf{x}_{B_{12}}^K - \mathbf{x}_{B_{12}}^n - \mu^n \mathbf{\Phi}_{B_{12}}^T \mathbf{\Phi} \mathbf{r}^n - \mu^n \mathbf{\Phi}_{B_{12}}^T \tilde{\mathbf{e}} \right\|_2 \\
&\leq 2 \left\| \mathbf{r}_{B_{12}}^n - \mu^n \mathbf{\Phi}_{B_{12}}^T \mathbf{\Phi} \mathbf{r}^n \right\|_2 + 2 \left\| \mu^n \mathbf{\Phi}_{B_{12}}^T \tilde{\mathbf{e}} \right\|_2 \\
&\leq 2 \left\| (\mathbf{I} - \mu^n \mathbf{\Phi}_{B_{12}}^T \mathbf{\Phi}_{B_{12}}) \mathbf{r}_{B_{12}}^n \right\|_2 \\
&\quad + 2 \left\| (\mu^n \mathbf{\Phi}_{B_{12}}^T \mathbf{\Phi}_{B_3}) \mathbf{r}_{B_3}^n \right\|_2 \\
&\quad + 2 \left\| \mu^n \mathbf{\Phi}_{B_{12}}^T \mathbf{\Phi} (\mathbf{x} - \mathbf{x}^K) \right\|_2 + 2 \left\| \mu^n \mathbf{\Phi}_{B_{12}}^T \mathbf{e} \right\|_2
\end{aligned}
$$

Splitting the set $B_{12}$ into two disjoint sets $B_1$ and $B_2$ with no more than $K$ elements each, we have by orthogonality and lemmata 7 and 8

$$
\begin{aligned}
\left\| \mu^n \mathbf{\Phi}_{B_{12}}^T \mathbf{\Phi}_{B_3} \mathbf{r}_{B_3}^n \right\|_2 &\\
&= \sqrt{ \left\| \mu^n \mathbf{\Phi}_{B_1}^T \mathbf{\Phi}_{B_3} \mathbf{r}_{B_3}^n \right\|_2^2 + \left\| \mu^n \mathbf{\Phi}_{B_2}^T \mathbf{\Phi}_{B_3} \mathbf{r}_{B_3}^n \right\|_2^2 } \\
&\leq \sqrt{2} \gamma_{2K} \left\| \mathbf{r}_{B_3}^n \right\|_2
\end{aligned}
$$

and

$$
\begin{aligned}
\left\| \mu^n \mathbf{\Phi}_{B_{12}}^T \mathbf{\Phi} (\mathbf{x} - \mathbf{x}^K) \right\|_2 &\\
&= \sqrt{ \left\| \mu^n \mathbf{\Phi}_{B_1}^T \mathbf{\Phi} (\mathbf{x} - \mathbf{x}^K) \right\|_2^2 + \left\| \mu^n \mathbf{\Phi}_{B_2}^T \mathbf{\Phi} (\mathbf{x} - \mathbf{x}^K) \right\|_2^2 } \\
&\leq \sqrt{2} (\gamma_{2K} + 1) \left[ \|\mathbf{x}_{T_1}\|_2 + \frac{\sum_{i=1}^L \|\mathbf{x}_{T_i}\|_1}{\sqrt{K}} \right].
\end{aligned}
$$

We now use the bounds in Lemma 7

$$
\begin{aligned}
\|\mathbf{r}^{n+1}\|_2 &\\
&\leq 2\sqrt{2} \gamma_{2K} \left( \left\| \mathbf{r}_{B_{12}}^n \right\|_2 + \left\| \mathbf{r}_{B_3}^n \right\|_2 \right) \\
&\quad + 2 \frac{\beta_{2K}}{\alpha_{2K}^2} \|\mathbf{e}\|_2 \\
&\quad + 2\sqrt{2}(\gamma_{2K} + 1) \left[ \|\mathbf{x} - \mathbf{x}^K\|_2 + \frac{\|\mathbf{x} - \mathbf{x}^K\|_1}{\sqrt{K}} \right]
\end{aligned}
$$

We also know that $\mathbf{r}_{B_{12}}^n$ and $\mathbf{r}_{B_3}^n$ are orthogonal so that

$$
\begin{aligned}
\|\mathbf{r}^{n+1}\|_2 &\leq 4\gamma_{2K} \|\mathbf{r}^n\|_2 + 2\frac{\beta_{2K}}{\alpha_{2K}^2} \|\mathbf{e}\|_2 \\
&\quad + 2\sqrt{2}(\gamma_{2K} + 1) \left[ \|\mathbf{x} - \mathbf{x}^K\|_2 + \frac{\|\mathbf{x} - \mathbf{x}^K\|_1}{\sqrt{K}} \right].
\end{aligned}
$$

The theorem follows by using $\gamma_{2K} < 1/8$

$$
\begin{aligned}
\|\mathbf{r}^{n+1}\|_2 &< 0.5 \|\mathbf{r}^n\|_2 + \frac{2.25}{\beta_{2K}} \|\mathbf{e}\|_2 \\
&\quad + 3.2 \left[ \|\mathbf{x} - \mathbf{x}^K\|_2 + \frac{\|\mathbf{x} - \mathbf{x}^K\|_1}{\sqrt{K}} \right]
\end{aligned}
$$

and iterating

$$
\begin{aligned}
\|\mathbf{r}^{n+1}\|_2 &< 2^{-n} \|\mathbf{x}^K\|_2 + \frac{5}{\beta_{2K}} \|\mathbf{e}\|_2 \\
&\quad + 6.4 \left[ \|\mathbf{x} - \mathbf{x}^K\|_2 + \frac{\|\mathbf{x} - \mathbf{x}^K\|_1}{\sqrt{K}} \right].
\end{aligned}
$$

## REFERENCES

[1] T. Blumensath and M. Davies, "How to use the iterative hard thresholding algorithm," in *Proc. SPARS'09*, 2009.

[2] M. Davies and N. Mitianoudis, "A simple mixture model for sparse overcomplete ICA," *IEE Proc.-Vision, Image Signal Process.*, vol. 151, no. 1, pp. 35–43, Aug. 2004.

[3] C. Fevotte and S. Godsill, "Sparse linear regression in unions of bases via bayesian variable selection," *IEEE Signal Process. Lett.*, vol. 13, no. 7, pp. 441–444, Jul. 2006.

[4] V. K. Goyal, M. Vetterli, and N. T. Thao, "Quantized overcomplete expansions in $R^N$: Analysis, synthesis and algorithms," *IEEE Trans. Inf. Theory*, vol. 44, no. 1, pp. 16–31, Jan. 1998.

[5] T. Blumensath and M. Davies, "Sparse and shift-invariant representations of music," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 1, pp. 50–57, Jan. 2006.

[6] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information.," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[7] E. Candès, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Commun. Pure Appl. Math.*, vol. 59, no. 8, pp. 1207–1223, 2006.

[8] D. Needell and J. Tropp, "COSAMP: Iterative signal recovery from incomplete and inaccurate samples," *Appl. Comput. Harmonic Anal.*, vol. 26, pp. 301–321, 2008.

[9] W. Dai and O. Milenkovic, "Subspace pursuit for compressed sensing: Closing the gap between performance and complexity," *Appl. Comput. Harmonic Anal.*, vol. 55, no. 5, pp. 2230–2249, 2009.

[10] T. Blumensath and M. Davies, "Iterative hard thresholding for compressed sensing," Applied and Computational Harmonic Analysis 2009, to appear in.

[11] A. Cohen, W. Dahmen, and R. DeVore, "Instance optimal decoding by thresholding in compressed sensing," RWTH Aachen, Germany, IGMP Rep. 289.

[12] R. Baraniuk, V. Cevher, M. Duarte, and C. Hegde, "Model-based compressive sensing," arXiv:0808.3572v2, 2008.

[13] T. Blumensath and M. E. Davies, "A simple, efficient and near optimal algorithm for compressed sensing," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2009, pp. 3357–3360.

[14] T. Blumensath and M. E. Davies, "Iterative thresholding for sparse approximations," *J. Fourier Anal. Applicat.*, vol. 14, no. 5, pp. 629–654, 2008.

[15] N. G. Kingsbury and T. H. Reeves, "Iterative image coding with overcomplete complex wavelet transforms," in *Proc. Conf. Visual Commun. Image Process.*, 2003.

[16] E. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, Mar. 2008.

[17] E. J. Candès and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2004.

[18] G. H. Golub and F. Van Loan, *Matrix Computations*, 3rd ed. Baltimore, MD: Johns Hopkins Univ. Press, 1996.

[19] S. Foucart and M.-J. Lai, "Sparsest solutions of underdetermined linear systems via $\ell_q$ minimization for $0 < q <= 1$," *Appl. Comput. Harmonic Anal.*, vol. 26, no. 3, pp. 395–407, 2009.

[20] E. van den Berg and M. P. Friedlander, "Probing the Pareto frontier for basis pursuit solutions," *SIAM J. Sci. Comput.*, vol. 31, no. 2, pp. 890–912, 2008.

[21] Z.-P. Liang and P. C. Lauterbur, *Principles of Magnetic Resonance Imaging: A Signal Processing Perspective*. New York: Wiley Blackwell, 1999.

[22] T. Blumensath and M. Davies, "Gradient pursuits," *IEEE Trans. Signal Process.*, vol. 56, no. 6, pp. 2370–2382, Jun. 2008.

[23] E. Candès, "The restricted isometry property and its implications for compressed sensing," California Inst. of Technol., Pasadena, Tech. Rep., 2008.
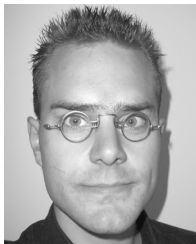
**Thomas Blumensath** (M'05) received the B.Sc. (honors) degree in music technology from Derby University, Derby, U.K., in 2002 and the Ph.D. degree in electronic engineering from Queen Mary, University of London, London, U.K., in 2006.

He was a Postdoctoral Research Fellow in the Institute for Digital Communication, University of Edinburgh, Edinburgh, U.K., and is now a Research Fellow in Applied Mathematics, University of Southampton, Southampton, U.K., working on inverse problems and sampling theory. He specializes in statistical and mathematical methods for signal processing with a particular focus on structured signal representations and their application to signal processing and acquisition problems.

**Mike Davies** (M'00) received the B.A. (honors) degree in engineering from Cambridge University, Cambridge, U.K., in 1989 and the Ph.D. degree in nonlinear dynamics and signal processing from University College London, London (UCL), U.K., in 1993.

He currently holds a SFC-funded chair in Signal and Image Processing, University of Edinburgh, Edinburgh, U.K. His current research interests include: sparse approximation, compressed sensing, and their applications.

Dr. Davies was awarded a Royal Society Research Fellowship in 1993 and was an Associate Editor for the IEEE TRANSACTIONS ON AUDIO , SPEECH, AND LANGUAGE PROCESSING (2003–2007).