RC24755 (W0903-015) March 4, 2009
Computer Science

IBM Research Report

Extended Compressed Sensing: Filtering Inspired Methods for Sparse Signal Recovery and Their Nonlinear Variants

Avishy Carmi
Department of Engineering
University of Cambridge
United Kingdom

Pini Gurfil
Technion - Israel Institute of Technology
Haifa 32000
Israel

Dimitri Kanevsky, Bhuvana Ramabhadran
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Cambridge - Haifa - India - T. J. Watson - Tokyo - Zurich

Extended Compressed Sensing: Filtering Inspired Methods for Sparse Signal Recovery and Their Nonlinear Variants

Avishy Carmi, *Member, IEEE*, Pini Gurfil, *Member, IEEE*, Dimitri Kanevsky and Bhuvana Ramabhadran

Abstract

New methods are presented for sparse signal recovery from a sequence of noisy observations. The sparse recovery problem, which is NP-hard in general, is addressed by resorting to convex and non-convex relaxations. The body of algorithms in this work extends and consolidate the recently introduced Kalman filtering (KF)-based compressed sensing methods. These simple methods, which are briefly reviewed here, rely on a pseudo-measurement trick for incorporating the norm relaxations following from CS theory. The extension of the methods to the nonlinear case is discussed and the notion of local CS is introduced. The essential idea is that CS can be applied for recovering sufficiently small and sparse state perturbations thereby improving nonlinear estimation in cases where the sensing function maps the state onto a lower-dimensional space. Other two methods are considered in this work. The extended Baum-Welch (EBW), a popular algorithm for discriminative training of speech models, is amended here for recovery of normalized sparse signals. This method naturally handles nonlinearities and therefore has a prominent advantage over the nonlinear extensions of the KF-based algorithms which rely on validity of linearization. The last method derived in this work is based on a Markov chain Monte Carlo (MCMC) mechanism. This method roughly divides the sparse recovery problem into two parts. Thus, the MCMC is used for optimizing the support of the signal while an auxiliary estimation algorithm yields the value of elements. An extensive numerical study is provided in which the methods are compared and analyzed. As part of this, the KF-based algorithm is applied to lossy speech compression.

I. Introduction

Recent studies have shown that sparse signals can be recovered accurately using less observations than what is considered necessary by the Nyquist/Shannon sampling principle; the emergent

Manuscript received ; revised .

A. Carmi is with the Signal Processing Group, Department of Engineering, University of Cambridge, UK.

P. Gurfil is with the Faculty of Aerospace Engineering, Technion – Israel Institute of Technology, Haifa 32000, Israel

D. Kanevsky and B. Ramabhadran are with IBM T. J. Watson Research Center, Yorktown, NY 10598, USA

theory that brought this insight into being is known as compressed sensing (CS) [1]–[3]. The essence of the new theory builds upon a new data acquisition formalism, in which compression plays a fundamental role. From a filtering standpoint, one can think about a procedure in which signal recovery and compression are carried out simultaneously, thereby reducing the amount of required observations. Sparse, and more generally, compressible signals arise naturally in many fields of science and engineering. A typical example is the reconstruction of images from under-sampled Fourier data as encountered in radiology, biomedical imaging and astronomy [4], [5]. Other applications consider model-reduction methods to enforce sparseness for preventing over-fitting and for reducing computational complexity and storage capacities. The reader is referred to the seminal work reported in [3] and [2] for an extensive overview of the CS theory.

The recovery of sparse signals is in general NP-hard [1], [6]. State of the art methods for addressing this optimization problem commonly utilize convex relaxations, non-convex local optimization and greedy search mechanisms. Convex relaxations are used in various methods such as LASSO [7], the Dantzig selector [8], basis pursuit and basis pursuit de-noising [9], and least angle regression [10]. Non-convex optimization approaches include Bayesian methodologies such as the relevance vector machine otherwise known as sparse Bayesian learning [11] as well as stochastic search algorithms which are mainly based on Markov chain Monte Carlo techniques [12]–[15]. Notable greedy search algorithms are the matching pursuit (MP) [16], the orthogonal MP [17], and the orthogonal least squares [18].

CS theory has drawn much attention to the convex relaxation methods. It has been shown that the convex l_1 relaxation yields an exact solution to the recovery problem provided two conditions are met: 1) the signal is sufficiently sparse, and 2) the sensing matrix obeys the so-called restricted isometry property (RIP) at a certain level. Another complementary result ensures high accuracy when dealing with noisy observations. Further elaboration of this result facilitated its probabilistic version which is concluded by the known statement of recovery 'with overwhelming probability'. To put it informally, it is highly probable for the convex l_1 relaxation to yield an exact solution provided the involved quantities, the sparseness degree s, and the sensing matrix dimensions $m \times n$ maintain relation of the type

$$s = \mathcal{O}(m/\log(n/m))$$

Influential as it may be, the theory of CS at its current stage deals with a parameter estimation problem in which the observations are merely a linear projection onto a lower dimensional space

$$y = Hx + \zeta, \qquad H \in \mathbb{R}^{m \times n}$$

In this work we are taking the underlying model two steps further, though not entirely from the theoretical standpoint. *Step 1:* The numerical recipes derived in this work are aimed at solving the discrete-time linear filtering problem where the noisy observation model assumes the above formulation. Here the time-varying signal is described via the state dynamics

$$x_{k+1} = Ax_k + w_k, \qquad k = 0, 1, 2, \ldots$$

It can be easily verified that if for instance $A = I$ (i.e., x_k is a random walk) then full reconstruction of the state x_k using the above measurement model is strictly infeasible. Nevertheless,

if the underlying signal is sparse in some basis then by introducing the known l_1 relaxation, accurate recovery is possible.

Step 2: Another extension which is of much interest involves the nonlinear counterpart of the above observation model

$$y = h(x) + \zeta, \qquad h : \mathbb{R}^n \rightarrow \mathbb{R}^m$$

In this work we present the notion of local RIP of the sensing function h which in turn facilitates the implementation of local CS. The idea here is that CS can be used to recover small and sparse perturbations Δx from a nominal state x^*. The l_1 relaxation then takes the form

$$\min \parallel \Delta x \parallel_1 \quad \text{s.t.} \quad \parallel y - h(x^* + \Delta x) \parallel_2 < \epsilon$$

which is approximately equivalent to

$$\min \parallel \Delta x \parallel_1 \quad \text{s.t.} \quad \parallel \bar{y} - [\partial h / \partial x]_{x^*} \Delta x \parallel_2 < \epsilon$$

where $\bar{y} = y - h(x^*)$. The approximate linear form above facilitates the application of the conventional CS where the estimation performance depends on the local RIP coefficient which is a property of the Jacobian $[\partial h / \partial x]$. This idea is further shown to improve the estimation performance of the extended Kalman filter when applied to the nonlinear observation model.

Algorithms: In broad, three types of algorithms are derived for solving the above mentioned problems. These account for 1) CS-embedded Kalman filter (CSKF) that was initially introduced in [19]–[21] and reviewed here for completeness, 2) Sparse Extended Baum-Welch (EBW), and 3) Stochastic subset search (S^3). We remark here that our work is not the first attempt to combine CS and KF. The Kalman filtering approach presented in [22] relies on an auxiliary optimization procedure (e. g., the Dantzig selector of [8]) and is capable of coping with time varying sparse signals. The method suggested inhere is based on a pseudo-measurement (PM) formulation of the underlying constrained optimization problem. Compared to the algorithm in [22], this method can be straightforwardly implemented in a stand-alone manner, as it is exclusively based on the well-known KF formulation.

1) The CSKF has three variants each of which is based on a different relaxation. The CSKF-1 uses the l_1 norm relaxation while the CSKF-p utilizes a quasi-norm formulation with l_p, $p \in (0, 1)$. The third variant is based on a unique l_0 norm approximation.
2) The sparse EBW method is based on a widely used algorithm in speech recognition. This method essentially maximizes a lower bound of a general objective function defined over a probability domain. The method is shown to naturally handle sparse directional vectors (i.e., $\parallel x \parallel_1 = 1$) and is guaranteed to converge.
3) The stochastic subset search is a Monte Carlo type method that uses both a simulated annealing core and a point process representations for finding the signal support. The estimated support is then fed to a conventional KF algorithm. This method is inspired by a Markov chain Monte Carlo scheme used for filtering of random finite sets.

Nonlinear Extensions:

1) A nonlinear variant of the CSKF is the CS-EKF. This algorithm utilizes the notion of local CS for recovering a sparse signal based on the aforementioned nonlinear observation model.
2) The sparse EBW naturally handles the nonlinear observation model and is essentially shown to outperform the CS-EKF in the normalized case (i.e., for $\| x \|_1 = 1$).

This paper is organized as follows. The next section mathematically formulates the sparse recovery problem. Section III provides a brief overview of the various CSKF algorithms. Local CS along with the nonlinear CS-EKF implementation are discussed in Section IV. The sparse EBW optimization method is introduced in Section V. Section describes the stochastic subset search method. Section VII provides the results of an extensive numerical study that had been carried out for assessing and comparing the various estimation methods. The last part of this section demonstrate the application of the CSKF to lossy speech compression. Finally, conclusions are offered in the last section.

II. Linear Estimation of Sparse Signals

Consider an $\mathbb{R}^n$-valued random discrete-time process $\{x_k\}_{k=1}^{\infty}$ that is sparse in some known orthonormal sparsity basis $\psi \in \mathbb{R}^{n\times n}$, that is

$$z_k = \psi^T x_k, \quad \#\{\mathrm{supp}(z_k)\} < n \tag{1}$$

where $\mathrm{supp}(z_k)$ and $\#$ denote the support of z_k and the cardinality of a set, respectively. Assume that z_k evolves according to

$$z_{k+1} = Az_k + w_k, \quad z_0 \sim \mathcal{N}(\mu_0, P_0) \tag{2}$$

where $A \in \mathbb{R}^{n\times n}$ and $\{w_k\}_{k=1}^{\infty}$ is a zero-mean white Gaussian sequence with covariance $Q_k \geq 0$. Note that (2) does not necessarily imply a change in the support of the signal. For example, A can be a block-diagonal matrix decomposed of A^d and A^n corresponding to the statistically independent elements $z^d \notin \mathrm{supp}(z_k)$ and $z^n \in \mathrm{supp}(z_k)$ where the respective noise covariance sub-matrices satisfy $Q^d = 0$ and $Q^n \geq 0$. The process x_k is measured by the $\mathbb{R}^m$-valued random process

$$y_k = Hx_k + \zeta_k = H'z_k + \zeta_k \tag{3}$$

where $\{\zeta_k\}_{k=1}^{\infty}$ is a zero-mean white Gaussian sequence with covariance $R_k > 0$, and $H := H'\psi^T \in \mathbb{R}^{m\times n}$.

Letting $y^k := [y_1, \ldots, y_k]$, our problem is defined as follows. We are interested in finding a y^k-measurable estimator, $\hat{x}_k$, that is optimal in some sense. Often, the sought after estimator is the one that minimize the mean square error (MSE) $E\left[\| x_k - \hat{x}_k \|_2^2\right]$. It is well-known that if the linear system (2), (3) is observable, i.e.,

$$\mathscr{O} := \begin{bmatrix} H^T & (HA)^T \cdots (HA^{n-1})^T \end{bmatrix}^T \quad \mathrm{rank}(\mathscr{O}) = n \tag{4}$$

then the solution to this problem can be obtained using Kalman filtering. On the other hand, if the system is unobservable, then the regular KF algorithm is useless; if, for instance, $A = I_{n\times n}$,

then it may seem hopeless to reconstruct x_k from an under-determined system in which $m < n$ and $\mathrm{rank}(H) < n$. Surprisingly, this problem may be circumvented by taking into account the fact that z_k is sparse.

A. The Combinatorial Problem and Compressed Sensing

Refs. [1], [6] have shown that in the deterministic case (i. e., when z is a parameter vector), one can accurately recover z (and therefore also x, i.e., $x = \psi z$) by solving the optimization problem

$$\min \parallel \hat{z} \parallel_0 \ \text{ s.t. } \sum_{i=1}^{k} \parallel y_i - H' \hat{z} \parallel_2^2 \leq \epsilon \tag{5}$$

for a sufficiently small ϵ, where $\parallel v \parallel_p = \left(\sum_{j=1}^{n} v_j^p \right)^{1/p}$ is the l_p-norm of v, and the zero-norm, $\parallel v \parallel_0$, is defined as[1] $\parallel v \parallel_0 := \#\{\mathrm{supp}(v)\}$.

Following a similar rationale, in the stochastic case the sought-after optimal estimator satisfies [2]

$$\min \parallel \hat{z}_k \parallel_0 \ \text{ s.t. } E_{z_k|y^k} \left[\parallel z_k - \hat{z}_k \parallel_2^2 \right] \leq \epsilon \tag{6}$$

Unfortunately, the above optimization problems are NP-hard and cannot be solved efficiently. Recently, it has been shown that if the sensing matrix H' obeys a so-called *restricted isometry property* (RIP) while z is sparse enough possibly with [2]

$$s = \mathcal{O}(m/\log(n/m)) \tag{7}$$

where $s = \#\{\mathrm{supp}(z)\}$, then the solution of the combinatorial problem (5) can almost always be obtained by solving the constrained convex optimization [1], [2]

$$\min \parallel \hat{z} \parallel_1 \ \text{ s.t. } \sum_{i=1}^{k} \parallel y_i - H' \hat{z} \parallel_2^2 \leq \epsilon \tag{8}$$

This is a fundamental result in the new emerging theory of compressed sensing (CS) [1], [2]. The main idea is that the convex l_1 minimization problem can be efficiently solved using a myriad of existing methods, such as LASSO [7], the Dantzig selector [8], Basis pursuit and Basis pursuit de-noising [9], and least angle regression [10], to mention only a few.

III. Kalman Filtering Compressed Sensing

It was only a matter of time until the Kalman filter, the work-horse of linear estimation theory, would be employed for compressed sensing. The popularity of the KF algorithm is owing to its ease of implementation and its modest computational demands (with respect to some other known estimation methods) as well as to its well-known statistical properties, such as being

[1]For $0 \leq p < 1$, $\parallel v \parallel_p$ is not a norm; the common terminology is *zero norm* for $p = 0$ and *quasi-norm* for $0 < p < 1$.

the best minimum MSE (MMSE) linear estimator around (which coincides with the optimal estimator in the MMSE sense for linear Gaussian systems) [23].

The first successful attempt for Kalman filtering-based compressed sensing was presented in [22] where the traditional KF algorithm was endowed with the Dantzig selector of [8]. This approach divides the sparse recovery problem into two interlaced subproblems: 1) support extraction, and 2) reduced order recovery. This separation roughly specifies a two phase algorithm in which some CS method (in this case the Dantzig selector) identifies the subset of elements in the support of the signal while the ordinary KF is applied for the reduced order system corresponding to the obtained subset. This approach, which is termed *interlaced CS* approach in this work, proved itself to be very successful as was demonstrated in [22].

A rather *straightforward* approach for solving the sparse filtering problem using KF was recently introduced in [19]–[21]. The suggested methods in both these works are based on a well-known trick for incorporating nonlinear equality constraints into the traditional KF formulation. Compared with the interlaced CS approach which involves an external optimization procedure, these methods are fairly simple to implement and requires no major modifications in the original KF structure. The emphasize here is on simplicity which makes these methods viable and appealing for many filtering applications. For completeness we revisit here the main concepts from [19]–[21].

1) Pseudo-Measurement Trick: For the system described by (2) and (3) the classical KF provides an estimate $\hat{z}_k$ that is a solution to the unconstrained l_2 minimization problem

$$\min_{\hat{z}_k} E_{z_k|y^k}\left[\| z_k - \hat{z}_k \|_2^2\right]$$

Inspired by the CS approach while retaining the KF objective function, we replace (6) by the constrained optimization

$$\min_{\hat{z}_k} E_{z_k|y^k}\left[\| z_k - \hat{z}_k \|_2^2\right] \text{ s.t. } \| \hat{z}_k \|_1 \leq \epsilon' \tag{9}$$

This procedure is based on the following proposition which is given here without a proof.

Proposition 1 ([21]): Let y and Y be an observation random variable and its realization, respectively. Let also x and $\hat{x}$ be a random variable and its associated y-measurable estimator, respectively. Given $y = Y$, the optimization problems

$$\min_{\hat{x}} E_{x|y}\left[\| x - \hat{x} \|_2^2\right] \quad \text{s. t.} \quad \|\hat{x}\|_1 \leq \epsilon_1 \tag{10}$$

$$\min_{\hat{x}} \|\hat{x}\|_1 \quad \text{s. t.} \quad E_{x|y}\left[\|x - \hat{x}\|_2^2\right] \leq \epsilon_2 \tag{11}$$

with ϵ_1, $\epsilon_2 > 0$, are equivalent.

The constrained optimization problem (9) can be solved in the framework of Kalman filtering using the pseudo-measurement (PM) technique [24], [25]. The idea is fairly simple: the inequality constraint $\| z_k \|_1 \leq \epsilon'$ is incorporated into the filtering process using a fictitious measurement $0 = \| z_k \|_1 - \epsilon'$, where ϵ' serves as a measurement noise. This PM can be rewritten as

$$0 = \bar{H} z_k - \epsilon', \quad \bar{H} := [\text{sign}(z_k(1)), \ldots, \text{sign}(z_k(n))] \tag{12}$$

where $\text{sign}(z_k(i))$ denotes the sign function of the ith element of z_k (i.e., $\text{sign}(z_k(i)) = 1$ if $z_k(i) > 0$ and equals 0 otherwise) . In this setting, the covariance R_ϵ of ϵ' is regarded as a tuning parameter, which can be determined based on simulation runs. A single iteration of the CS-embedded KF is detailed in Algorithm 1 [2].

Algorithm 1 CSKF-1 [21]

1: *Prediction*

$$\hat{z}_{k+1|k} = A\hat{z}_{k|k} \tag{13a}$$

$$P_{k+1|k} = AP_{k|k}A^T + Q_k \tag{13b}$$

2: *Measurement Update*

$$K_k = P_{k+1|k}H'^T \left(H'P_{k+1|k}H'^T + R_k\right)^{-1} \tag{14a}$$

$$\hat{z}_{k+1|k+1} = \hat{z}_{k+1|k} + K_k \left(y_k - H'\hat{z}_{k+1|k}\right) \tag{14b}$$

$$P_{k+1|k+1} = (I - K_kH')P_{k+1|k} \tag{14c}$$

3: *CS Pseudo Measurement:* Let $P^1 = P_{k+1|k+1}$ and $\hat{z}^1 = \hat{z}_{k+1|k+1}$.

4: **for** $\tau = 1, 2, \ldots, N_\tau - 1$ iterations **do**

5:

$$\bar{H}_\tau = [\text{sign}(\hat{z}^\tau(1)), \ldots, \text{sign}(\hat{z}^\tau(n))] \tag{15a}$$

$$K^\tau = P^\tau \bar{H}_\tau^T \left(\bar{H}_\tau P^\tau \bar{H}_\tau^T + R_\epsilon\right)^{-1} \tag{15b}$$

$$\hat{z}^{\tau+1} = (I - K^\tau \bar{H}_\tau)\hat{z}^\tau \tag{15c}$$

$$P^{\tau+1} = (I - K^\tau \bar{H}_\tau)P^\tau \tag{15d}$$

6: **end for**

7: Set $P_{k+1|k+1} = P^{N_\tau}$ and $\hat{z}_{k+1|k+1} = \hat{z}^{N_\tau}$.

2) Quasi-Norm Constrained Variants: A different approach for approximately solving the combinatorial problem in (6) is based on replacing $\|\cdot\|_0$ by a quasi-norm $\|\cdot\|_p$ with $0 < p < 1$. This approach has already been shown to yield better accuracy compared to the l_1 norm [6].

Following the previous section methodology, the PM technique is used here to incorporate the quasi-norm inequality constraint $\| z_k \|_p \leq \epsilon'$ by producing the fictitious measurement

$$0 = \| z_k \|_p - \epsilon'$$

where ϵ' serves as a zero-mean Gaussian measurement noise with covariance R_ϵ. In practice, this PM is linearized around some nominal state z_k^* to yield

$$0 = \left(\sum_{i=1}^{n} |z_k^*(i)|^p\right)^{1/p} + \bar{H}\Delta z_k - \epsilon' + \mathcal{O}(\| \Delta z_k \|_2^2) \tag{16}$$

[2]Notice that this is an unusual implementation of the KF as the matrix $\bar{H}_\tau$ is state dependent.

where $z_k(i)$ denotes the ith element of z_k, the perturbation $\Delta z_k := z_k - z_k^*$, and

$$\bar{H}(i) = \begin{cases} \left(\sum_{i=1}^{n} |z_k^*(i)|^p\right)^{1/p-1} [z_k^*(i)]^{p-1}, & \text{if } z_k^*(i) > 0 \\ -\left(\sum_{i=1}^{n} |z_k^*(i)|^p\right)^{1/p-1} [-z_k^*(i)]^{p-1}, & \text{if } z_k^*(i) \leq 0 \end{cases}, \qquad i = 1, \ldots, n \tag{17}$$

is the ith element of $\bar{H}$. This formulation facilitates the implementation of an extended KF (EKF) stage for incorporating the PM. Following this, the nominal state z_k^* is set as the updated estimate at time k.

A single iteration of the resulting KF algorithm with the linearized PM stage is similar to Algorithm 1 with a slight modification in the PM implementation as described in Algorithm 2.

Algorithm 2 PM Stage of The CSKF-p [21]

1: *Pseudo Measurement:* Let $P^1 = P_{k+1|k+1}$ and $\hat{z}^1 = \hat{z}_{k+1|k+1}$.
2: **for** $\tau = 1, 2, \ldots, N_\tau - 1$ iterations **do**
3: Compute $\bar{H}_\tau$ using (17) with $z_k^* = \hat{z}^\tau$.

$$K^\tau = P^\tau \bar{H}_\tau^T \left(\bar{H}_\tau P^\tau \bar{H}_\tau^T + R_\epsilon\right)^{-1} \tag{18a}$$

$$\hat{z}^{\tau+1} = \hat{z}^\tau - K^\tau \parallel \hat{z}^\tau \parallel_p \tag{18b}$$

$$P^{\tau+1} = (I - K^\tau \bar{H}_\tau) P^\tau \tag{18c}$$

4: **end for**
5: Set $P_{k+1|k+1} = P^{N_\tau}$ and $\hat{z}_{k+1|k+1} = \hat{z}^{N_\tau}$.

3) Approximate l_0 Norm: The l_0 norm can alternatively be approximated by

$$n - \sum_{i=1}^{n} \exp\left(-\alpha |z_k(i)|\right) \tag{19}$$

for large enough $\alpha > 0$. The corresponding PM stage in that case consists of the same steps (18) where (18b) is replaced by

$$\hat{z}^{\tau+1} = \hat{z}^\tau + K^\tau \left[n - \sum_{i=1}^{n} \exp\left(-\alpha |\hat{z}^\tau(i)|\right) \right] \tag{20}$$

(i.e., the PM is $n = \sum_{i=1}^{n} \exp\left(-\alpha |z_k(i)|\right) + \epsilon'$) where $\bar{H}$ is given by

$$\bar{H}(i) = \begin{cases} -\alpha \exp\left(-\alpha z_k^*(i)\right), & \text{if } z_k^*(i) > 0 \\ \alpha \exp\left(\alpha z_k^*(i)\right), & \text{if } z_k^*(i) \leq 0 \end{cases}, \qquad i = 1, \ldots, n \tag{21}$$

IV. Extended Compressed Sensing

At its current stage the theory of CS deals with the recovery of signals that are linearly projected onto a lower dimension observation space. One could naturally wonder whether a similar set of rules apply in the case of arbitrary smooth mappings. The formulation would then

be the following. Given a sufficiently smooth mapping $h : \mathbb{R}^n \to \mathbb{R}^m$, $m < n$ and some s-sparse vector $z \in \mathbb{R}^n$ that obey the observational relation $y_i = h(z) + \zeta, \quad i = 1, \ldots, k$, then to what extent and under what conditions can we recover z from y using the l_1 relaxation suggested by CS, i.e., by solving

$$\min \parallel \hat{z} \parallel_1 \ \text{ s.t. } \sum_{i=1}^{k} \parallel y_i - h(\hat{z}) \parallel_2^2 \leq \epsilon \tag{22}$$

It should be mention that such a nonlinear observation model was recently addressed in [26], though from a greedy standpoint, i.e., by generalizing the orthogonal matching pursuit algorithm. In this work we are not going to fully answer the above stated question but rather demonstrate how CS can be applied for recovering sufficiently small sparse perturbations from a given nominal state.

One of the fundamental results in CS is that accurate and possibly exact recovery of sparse signals is feasible depending on the RIP level of the sensing matrix [2]. The RIP is closely related to the the Johnson-Lindenstrauss (JL) lemma which is stated about general Lipschitz low-distortion embeddings [27].

Lemma 1 (JL): Given some $\delta \in (0, 1)$, a set $\mathcal{Z}$ of l points in $\mathbb{R}^n$ and a number $m_0 = \mathcal{O}(\ln(l)/\delta^2)$, there is a Lipschitz function $h : \mathbb{R}^n \to \mathbb{R}^m$ where $m > m_0$ such that

$$(1 - \delta) \parallel z - z^* \parallel_2 \leq \parallel h(z) - h(z^*) \parallel_2 \leq (1 + \delta) \parallel z - z^* \parallel_2 \tag{23}$$

for all $z, z^* \in \mathcal{Z}$.

Now, consider a case where $z = z^* + \Delta z$ with sufficiently small $\parallel \Delta z \parallel_2$, then by taking the first-order Taylor expansion of $h(z)$ around z^* it can be easily recognized that the JL lemma reduces to approximately the RIP of the Jacobian $[\partial h/\partial z]$ computed locally at z^*, that is

$$(1 - \delta) \parallel \Delta z \parallel_2 \leq \parallel [\partial h/\partial z]_{z^*} \Delta z + o\left(\parallel \Delta z \parallel_2^2\right) \parallel_2 \leq (1 + \delta) \parallel \Delta z \parallel_2 \tag{24}$$

In that sense the Lipschitz function that satisfies the JL relation (23) *locally* obeys the RIP at z^* for the perturbation vector Δz. The property (24) of the sensing function $h(z)$ is termed *Local RIP* in this work. Similarly to the linear case, the level of the local RIP of $h(z)$ at z^* is determined according to the maximal sparseness degree s of the perturbation Δz for which (24) holds. Obviously, when considering the recovery of a sufficiently small and sparse Δz, CS can be applied where the Jacobian $[\partial h/\partial z]_{z^*}$ takes the role of the traditional sensing matrix. The l_1 relaxation would then have the form

$$\min \parallel \Delta \hat{z} \parallel_1 \ \text{ s.t. } \sum_{i=1}^{k} \parallel y_i - h(z^* + \Delta \hat{z}) \parallel_2^2 \leq \epsilon \tag{25}$$

where the accuracy of recovery would be related to the local RIP constant δ_s of the sensing function $h(z)$.

A. Local CS and Nonlinear Estimation: The CS-Embedded EKF

Maybe the most interesting implication that follows from the local CS idea is that the l_1 relaxation can improve conventional nonlinear estimation methods that are based on linearization such as the EKF. In such methods the linearization around some predetermined nominal point, which is usually taken as the best up to date estimate, facilitates the application of a linear estimator (e.g., the KF) for reconstructing the perturbed state. The sensing matrix in this case is merely the Jacobian of the sensing function locally computed at the nominal point whereas the state transition matrix is taken as the Jacobian of the time propagation function. Now, consider a case in which the sensing function $h(z)$ maps the state onto a lower-dimensional space, then following the preceding argument it is expected that local CS will allow better recovery of sufficiently sparse perturbation Δz provided that $h(z)$ obeys the local RIP at a proper level.

In this work we have implemented the local CS idea by amending the CSKF algorithms of Section III for nonlinear estimation. The slight modification consists of replacing the ordinary KF recursion with an EKF one while retaining the desired PM stage (i.e., corresponding to the l_1, the l_p or the approximate l_0 norms).

V. Sparse Optimization: The Extended Baum-Welch

Extended Baum-Welch (EBW) is a popular optimization technique in speech recognition to optimize discriminative objective functions [28]. As the name suggests, EBW is an extension of the BW algorithm. The Baum-Welch (BW) algorithm is an expectation-maximization algorithm that computes maximum likelihood estimates and posterior mode estimates for the parameters (transition and emission probabilities) of a Hidden Markov Model, when given only emissions as training data [29]. The BW algorithm can be used to optimize non-negative polynomials over probability domains. More precisely it can be used to solve the following problem:

$$\max_{z} Q(z) \quad \text{s.t. } z \in P = \{z_{ij} \geq 0, \quad \sum_{j=1}^{m_i} z_{ij} = 1\} \tag{26}$$

and Q is a polynomial with non-negative coefficients and P is a discrete probability domain. This is an iterative procedure that uses the Jensen inequality to reduce the optimization for each recursive step to optimization of an auxiliary function over P. The auxiliary function is estimated at each step and is a weighted sum of $\log z_{ij}$. In practical tasks that require maximum-likelihood estimates of HMM with large number of parameters the BW method became popular because it is easy to implement, it usually requires a small number of iterations to get to near optimum and each iteration requires number of computations that is proportional to a number of parameters. The BW algorithm was not applicable directly to discriminative estimation problems for HMM that required optimization of (26) where Q was a polynomial with some negative coefficients since the Jensen inequality could not be applied there. Twenty years ago the following simple trick was found how to extend BW methods to polynomials that contain negative coefficients [28]. It was observed that an optimization problem of maximzing Q does not change if ones adds to Q a polynomial $W(z) = D(\sum_{ij} z_{ij} + 1)^m$ that is a constant in the

probability domain P. In other words $Q(z)+W(z)=Q(z)+D\times L$ where $L=(\sum_{ij}z_{ij}+1)^m$ is a constant in P and $\arg\max_z Q(z)=\arg\max_z\{Q(z)+W(z)\}=\arg\max_z\{Q(z)+D\times L\}$. Now if one takes m a degree of the polynomial Q and sufficiently large D then all coefficients of the polynomial $Q(z)+W(z)$ have non-negative coefficients and therefore the Jensen inequality and as a consequence the BW procedure is applicable to it. The extension of BW to polynomials that contain negative coefficients immediately allows the extention of BW to rational functions over probability domains. Then via a suitable approximation process BW has also been extended to discriminative functions defined over sets of Gaussian distributions rather than discrete probabilities. EBW has become a popular method in discriminative speech recognition tasks in the last decade because of its ease of implementation and convergence properties.

In this Section we show EBW recursion can be naturally applied to maximizing a differentiable function over a domain consisting of parameters whose q-norms equal 1. It was explained in previous sections that 1-norm constraints lead to compressive sensing conditions. Therefore we call EBW methods for optimizing functions over 1-norm constraints as sparse optimization.

The following theorem [30] is needed to extend EBW methods to sparse constraints.

Theorem 1: Let $F(z)$ be a function that is defined over $P=\{z_{ij}\geq 0,\sum_j z_{ij}=\sum_{j=1}^{j=m_i}z_{ij}=1\}$. Let F be differentiable at $z\in P$. Let $c_{ij}=z_{ij}\frac{\partial}{\partial z_{ij}}F(z)$, and let $\hat{z}=\{\hat{z}_{ij}\}\neq z=\{z_{ij}\}$ where

$$\hat{z}_{ij}=\frac{c_{ij}+z_{ij}D}{\sum_i c_{ij}+D} \tag{27}$$

Then $F(\hat{z})>F(z)$ for sufficiently large positive D and $F(\hat{z})<F(z)$ for sufficiently small negative D.

Remark: This theorem was proved in [28] for rational functions. If $F(z)$ is a polynomial with non-negative coefficients then $F(\hat{z})>F(z)$ for $D=0$ and EBW coincides with BW. This is a special case of the fact that the ML estimation of HMM parameters via EBW coincides with the BW.

For improved reading the proof of Theorem 1 is deferred to the Appendix.

EBW Over Fractional Norms

Let $Q(y)$ be a differentiable function of $y=\{y_i\}\in\mathbb{R}^n, i=1,...n$. Let us consider the following problem:

$$\max_y Q(y)\quad \text{s.t. } \parallel y\parallel_q\leq\beta \tag{28}$$

We solve this problem by transforming (28) into a problem over a probability domain for which EBW update rules (27) exist. Let us consider the l_1 norm, that is by setting $y_i=x_i^{1/q}$, $F(\{x_i\})=Q(\{y_i\})$ and $\epsilon=\beta^q$. The problem (28) then becomes

$$\max_x F(x)\quad \text{s.t. } \parallel x\parallel_1\leq\epsilon \tag{29}$$

Now, using the dummy variable $x_0\geq 0$ the above problem is rewritten as

$$\max_x F(x)\quad \text{s.t. } \parallel x\parallel_1+x_0=\epsilon \tag{30}$$

Further letting $v_i = x_i/\epsilon$, $i = 0, ...n$, and $F(x) = F(\{\epsilon v_i\}) = G(v)$ we may write

$$\max_v G(v) \quad \text{s.t. } \| v \|_1 = 1 \tag{31}$$

Recognizing that

$$\| v \|_1 = \sum_i \sigma(v_i) v_i \tag{32}$$

and

$$G(v) = G(\{\sigma(v_i)\sigma(v_i)v_i\}) = G(\sigma(v_i)z_i) = G(z) \tag{33}$$

where $\sigma(v_i) = \text{sign}(v_i)$, and $z = \{z_i\} = \{\sigma(v_i)v_i\}$ allows writing an equivalent problem to (29) which takes the form

$$\max_z G(z) \quad \text{s.t. } \sum z_i = 1,\ z_i \geq 0 \tag{34}$$

Problems of the type (34) have an EBW based solutions that can be obtained by iterating (27). The detailed EBW recursion for solving (29) is given in Algorithm 3.

Algorithm 3 Sparse Directional EBW

1: Set intial conditions x^0 $s.t.$ $\| x^0 \|_1 = 1$
2: **for** $t = 0, 1, \ldots, N_t$ iterations **do**
3: Set $z_i^t = \sigma(x_i^t)x_i^t$
4: $G_t(z^t) = F(\{\sigma(x_i^t)\sigma(x_i^t)x_i^t\}) = F(\sigma(x_i^t)z_i^t)$
5: Compute coefficents

$$c_j = c_j(z^t) = \frac{\partial G_t(z^t)}{\partial z_j^t}$$

6: Adapt D according to some rule, e.g., $D^* = \arg\max_D G_t(\{z_i^{t+1}(D)\})$
7: Update estimate

$$z_i^{t+1} = z_i^{t+1}(D) = \frac{(c_i + D)z_i^t}{\sum_j c_j z_j^t + D}$$

8: **end for**

A. Tuning D

Note that the sign of z_i^{t+1} in Algorithm 3 depends on how large is D. Namely, it is positive for sufficiently large D and is negative if $c_i + D < 0$. In practice, instead of computation of $D^* = \arg\max_D G_t\{z_i^{t+1}(D)\}$ one can use various approximate schemes. As a general rule D should increase significantly when a local maximum of an objective function is approached. One way to achieve this is to choose D that is inversely proportional to some degree of a gradient to an objective function at a point that is being updated during an iterative optimization process. Various gradient steepness metrics that could be used for tuning D for EBW update rules for Gaussian parameters are described in [31]. Several popular strategies for tuning D in speech recognition tasks are introduced in [32].

VI. Stochastic Subset Search

In recent years, Monte Carlo (MC) methods and in particular Markov chain MC (MCMC) have been successfully implemented for vast high-dimensional optimization and filtering applications. Their popularity is a direct consequence of their flexibility, their problem solving capabilities and the ever increasing processing power of todays computers. Being a simulation based approach, MCMC generally imposes no restrictions on the characteristics of the problem being solved. In addition, smart strategies that have been developed over the past years improved the efficiency of these methods when dealing with multi-modality and varying dimensionality [33]. The reader is referred to [34]–[37] for extensive overview and applications of MCMC.

In this section we derive a MCMC-based sparse recovery algorithm. The MCMC approach inhere is inspired by the particles algorithm used in [38] for multi-target tracking. However, in this work the MCMC particles mechanism is used for optimization rather than for filtering. The formal derivation proceeds as follows.

A. Random Set Representations

We exploit the following formulation which is used for representing random finite sets. Consider a $\mathbb{R}^n$-valued indicators random vector e of which each element may take either values 0 or 1. Having $e^i = 1$ implies that the ith element is active. The indicators vector is associated with a vector $z = [z^1, \ldots, z^n]^T$. Both these quantities represent a random set S, that is

$$S = \{z^i \mid \quad e^i = 1, \quad i = 1, \ldots, n\} \tag{35}$$

In the context of our sparse recovery problem the vector e represents the unknown support of the signal and is essentially optimized using a MCMC mechanism whereas the corresponding values in S are obtained using a traditional KF. In that sense, this technique follows the *interlaced CS* concept discussed in Section III.

B. MCMC Particles

Suppose that at every time step k we have N candidates (particles) $\{e_k(j), z_k(j)\}_{j=1}^N$. Each particle represents a set of dimension $\sum_i e_k^i(j)$ that correspond to a subset of the sensing matrix H'. Let us denote $H(j)$ the subset corresponding to the jth particle, that is, $H(j) = \{H_i' \mid e_k^i(j) = 1, \quad i = 1, \ldots, n\}$ where H_i' denotes the ith column of H'. We define for each particle a scoring function of the form

$$L(S(j)) = \exp\left\{-\frac{1}{2}(1-\gamma_k)\left(y_k - H(j)\hat{S}(j)\right)^T V_k(j)^{-1}\left(y_k - H(j)\hat{S}(j)\right)\right\} \tag{36}$$

where the jth set $\hat{S}(j)$ is taken as the output of an auxiliary estimator that was applied for processing y_k with an initial state $S(j)$ and a sensing matrix $H(j)$. As it would become clear in the ensuing, the time-dependent scaling parameters $\gamma_k \in [0, 1]$ and $V_k(j) \in \mathbb{R}^{m \times m}$ affect the

sampling efficiency of the method. In this work we have used a KF for obtaining $\hat{S}(j)$ where V_k is set accordingly as the innovations covariance, that is

$$V_k = H(j)P_k H(j)^T + R_k \tag{37}$$

where P_k is the corresponding KF covariance. At this point we use the Metropolis-Hastings (MH) algorithm for producing an improved particles population. Thus, every new candidate $S(j)'$, $j = 1, \ldots, N$ is accepted with probability of $L(S(j)')/L$ where L denotes the scoring function of the previously accepted one. The obtained population serves as the initial set of particles at the next time step $\{e_{k+1}(j), z_{k+1}(j)\}_{j=1}^{N}$. The estimated signal $\{e_k^*, z_k^*\}$ is then taken as the one having the maximal acceptance rate.

Similarly to the cooling schedule in simulated annealing, here, the 'tempering' parameter γ_k is used for regulating the algorithm's convergence rate. The ability of the algorithm to overcome local maxima traps greatly depends on a good choice of the cooling scheme. Following a thumb-rule from MH theory, a fairly good tempering schedule allows accepting between 20% to 40% of the purposed candidates [37].

C. Birth/Death Moves

A good exploration of the search space is maintained by incorporating birth/death type moves. In this work the indicators e_k^i, $i = 1, \ldots, n$ are assumed to evolve according to a Markov chain with the transition kernel

$$p(e_k^i \mid e_{k-1}^i = j) = \begin{cases} a_j, & \text{if } e_k^i = j \\ 1 - a_j, & \text{otherwise} \end{cases} \tag{38}$$

where a_j denotes the probability of staying in state $j \in [0, 1]$.

VII. Numerical Study

In this section we demonstrate the performance of the derived algorithms as well as some additional concepts that were introduced in previous sections. A major part here is devoted to the comparison of the various algorithms in different cases. Thus, the CSKF variants (i.e., the l_1, the l_p and the approximate l_0 norms) are compared with the sparse EBW implementation as well as with the S^3 method. We demonstrate the performance of the CSKF algorithms both in the static and dynamic cases. We then proceed on with the nonlinear implementations, the CS-EKF endowed with the various norms that are compared with the sparse EBW method. In addition, example is given that exemplifies the implementation of the S^3 method for finding the RIP coefficient of an arbitrary matrix. The last part of this section demonstrates the application of the CSKF for lossy speech compression.

A. Static Case

The following example is partially based on the one in [19]–[21]. Here the signal $z \in \mathbb{R}^{256}$ is assumed to be a sparse parameter vector (i.e., $A = I_{256\times 256}$, $Q_k = 0$). The signal support consists of total of 10 elements $z(i) \neq 0$ of which both the index and value are uniformly sampled over $i \sim U_i[1, 256]$ [3] and $z(i) \sim U[-10, 10]$, respectively. The sensitivity matrix H' consists of 72 rows in which the elements are sampled from a Gaussian distribution $\mathcal{N}(0, 1/72)$. The columns of H' are normalized following the example in [22] (this matrix has been shown to satisfy the RIP at a sufficient level, see [2], [22]). The observation noise covariance is set as $R_k = 0.001^2 I_{72\times 72}$.

1) Algorithms Settings: The tuning covariance R_ϵ of the CSKF-p was set as 20000^2 and 200^2 for $p = 0.5$ and $p = 1$, respectively. The alternative l_0 approximation (19) is implemented using $\alpha = 1$ and $R_\epsilon = 100^2$ (these values were chosen based on tuning runs for achieving ideal performance in terms of accuracy).

The sparse EBW was implemented using the following objective function

$$G(z) = p(y \mid z) \propto \prod_{i=1}^{k} \exp\left\{-\frac{1}{2}\left(y_i - H'z\right)^T R_i^{-1}\left(y_i - H'z\right)\right\} \tag{39}$$

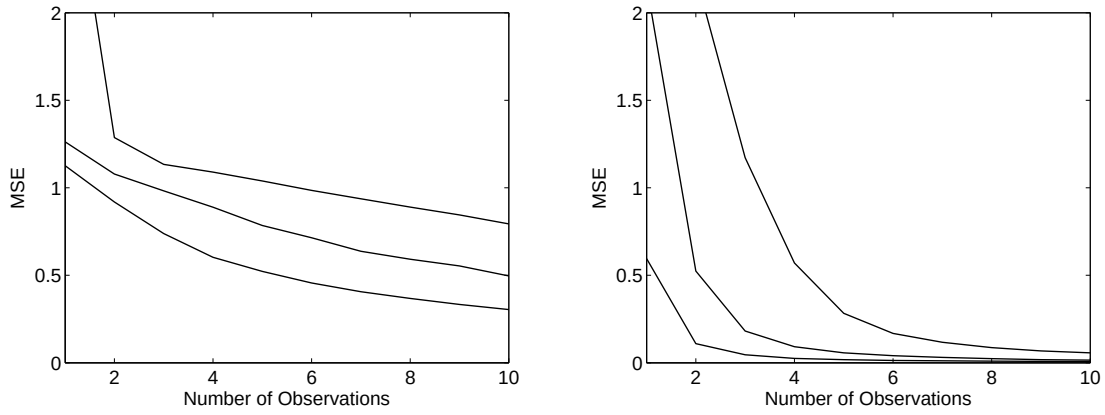
using not more than 10 iterations per time step. The estimation procedure was performed in a sequential fashion by taking the best estimate of the preceding time step as the initial state for the next one.

The S^3 algorithm used a total of 1000 particles. The cooling parameter was set as $\gamma_0 = 1-10^{-7}$ and was rapidly reduced at increments of 10^{-6}. The birth/death moves probability was set as $a_j = 0.9$ for $j = 0, 1$. These settings maintained an average of 30% acceptance rate of the MH.

2) Results: Before comparing the various methods we demonstrate the affect of some of the parameters on the estimation performance in this case [21]. Thus, Fig. 1 depicts the mean square estimation error based on 50 Monte Carlo runs for various number N_τ of PM iterations of the CSKF-p algorithm with $p = 0.5, 1$. As expected, increasing N_τ yields an improved estimation performance as the estimation error attains lower values. It can be clearly seen that the CSKF-0.5 outperforms the CSKF-1 using the same number of iterations. This fact is further demonstrated in the next few figures. In an additional figure, Fig. 2, a snapshot at time $k = 20$ shows the performance of both the CSKF-1 and an ordinary KF (i.e., without the PM stage) in a typical run. As it could be expected the ordinary KF implementation is useless as the underlying system is unobservable.

The estimation performance of the various methods in the static case is shown in Fig. 3. This figure, which consists of the mean square estimation error based on 100 Monte Carlo runs of the various methods, corresponds to two cases one of which involves an actual normalized signal $z/\parallel z \parallel_1$. The purpose of this is to allow a fair comparison with the EBW method that is intended for recovering normalized (or directional) signals. Thus, the only methods that are considered in Fig. 3a are the CSKF-p with $p = 0.5, 1$, the CSKF with the approximate l_0 norm, and the S^3

[3] $U_i[a, b]$ denotes a discrete uniform distribution of which the support are all the integers in the interval $[a, b]$.



(a) CSKF-1 **(b) CSKF-**0.5

Fig. 1: Mean square estimation error based on 50 runs of the CSKF-1 (a) and of the CSKF-0.5 (b) for various numbers of PM iterations. The lines, top to bottom, correspond to 50, 100, and 200 pseudo-measurement iterations. Static case. [19]–[21]

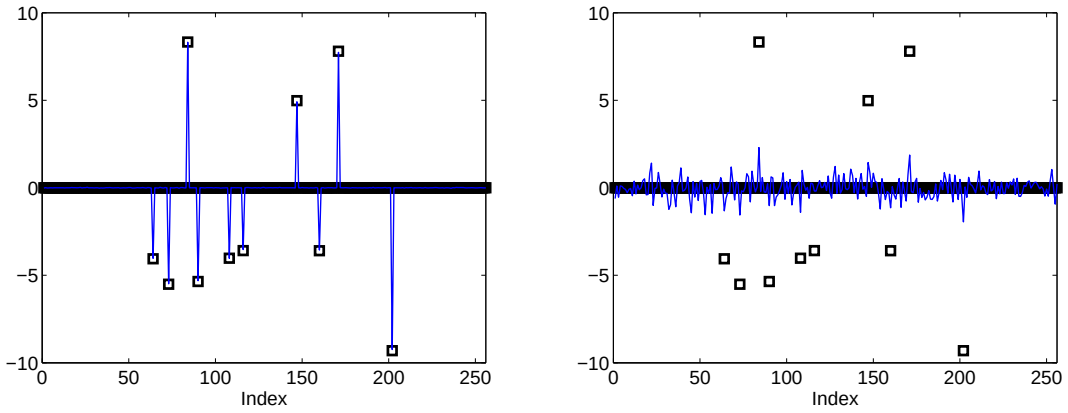
method. It can be clearly seen that the best estimation accuracy is attained while using either the quasi-norm l_p with $p = 0.5$ or the S^3 algorithm. Both these algorithm attain estimation accuracy at the level of the observation noise (i.e., 10^{-3}). The alternative l_0 approximation is slightly less accurate but tends to converge faster. The CSKF-1 Algorithm exhibits inferior estimation accuracy with respect to the other methods. The estimation performance of an ordinary KF that is aware of the signal support is shown to attain errors of approximately 0.9×10^{-3}, slightly better than the CSKF-0.5 and the S^3. Proceeding to Fig. 3b, the EBW exhibits inferior estimation performance compared to the CSKF-1 (and respectively with all other variants of the CSKF). The EBW version that allows the recovery of negative components is shown to perform worser than its non-negative counterpart mainly due to some runs at which the algorithm did not converge.

The 1σ bounds corresponding to $z(1) - \hat{z}(1)$ as computed by the various filters in a single run are shown in Fig. 4. It can be seen that these bounds reflect the accuracy when using the different norms in the CS stage. We have already seen that the best estimation performance was achieved by the CSKF-0.5 and correspondingly its 1σ bounds are the tightest.

B. Dynamic Case

The various CSKF algorithms are applied to filtering of sparse random-walk processes in [19]–[21]. The reader is, therefore, referred to these works for additional details and insights.

Here we have excluded both the EBW and S^3 methods for the following reasons. In its formulation presented inhere, the EBW is not suitable for the recovery of random processes. It should be noted that this issue is a part of the authors ongoing research. The implementation of



(a) CSKF-1 **(b) Regular KF**

Fig. 2: A snapshot at $k = 20$ in a typical run of the CSKF-1 (a) using $N_\tau = 200$ PM iterations and of an ordinary KF (b). Showing the elements of the actual (squares) and estimated (lines) signals. Static case.

the S^3 algorithm for this case is trivial and does not reflect the performance of the subset search mechanism.

C. Nonlinear Extensions

This part of the numerical study demonstrates the idea of extended CS (or local CS) of Section IV and its application to nonlinear estimation. We consider a recovery problem consisting of the following nonlinear observation model

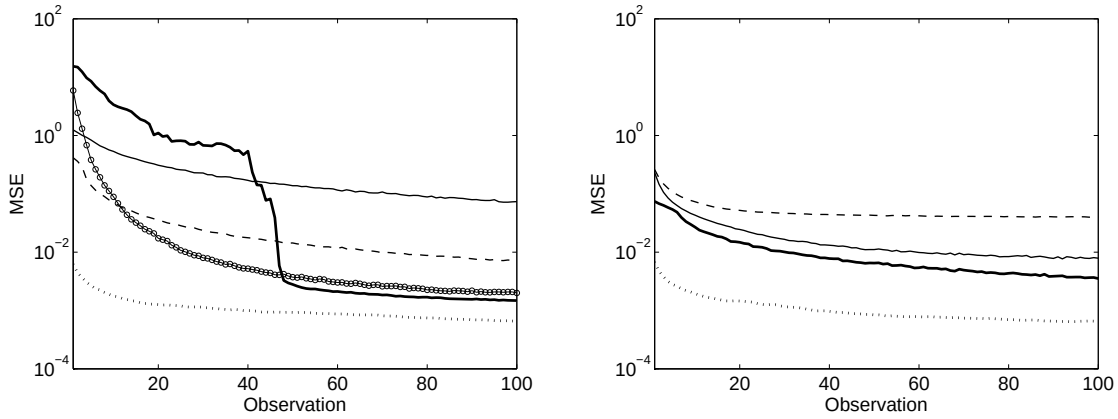
$$y_i = h(z) + \zeta_i, \quad h(z) = H' \left[\mathrm{diag}(z)^{1\frac{1}{2}} \mathbf{1} + az(j)\mathbf{1} \right] \tag{40}$$

where $\mathbf{1}$ denotes a vector of which all entries are 1's, a is some constant, and j is an arbitrary number between 1 and n. The Jacobian matrix of the sensing function $h(z)$ is given by

$$\frac{\partial h}{\partial z} = 1\frac{1}{2} H' \mathrm{diag}(z)^{\frac{1}{2}} + aH' \mathrm{diag}(\boldsymbol{e}_j) \tag{41}$$

where $\boldsymbol{e}_j \in \mathbb{R}^n$ has its jth entry equals one while all others are zero. Similarly to the previous examples, the random matrix $H' \in \mathbb{R}^{72\times 256}$ has its entries independently sampled from a zero-mean normal distribution with variance $1/72$. The vector z has 10 non-zero elements of which the locations are uniformly sampled over the integers in the interval $[1, 256]$. The values of the elements in the support of z are uniformly sampled between $[0.5, 1.5]$. All other algorithm and noise related parameters are set as before.

From the above it is evident that the local RIP of $h(z)$ is affected by the parameter a. Taking a too large may violate this desired property thereby deteriorating the attainable estimation



(a) Unnormalized z **(b) Normalized z**

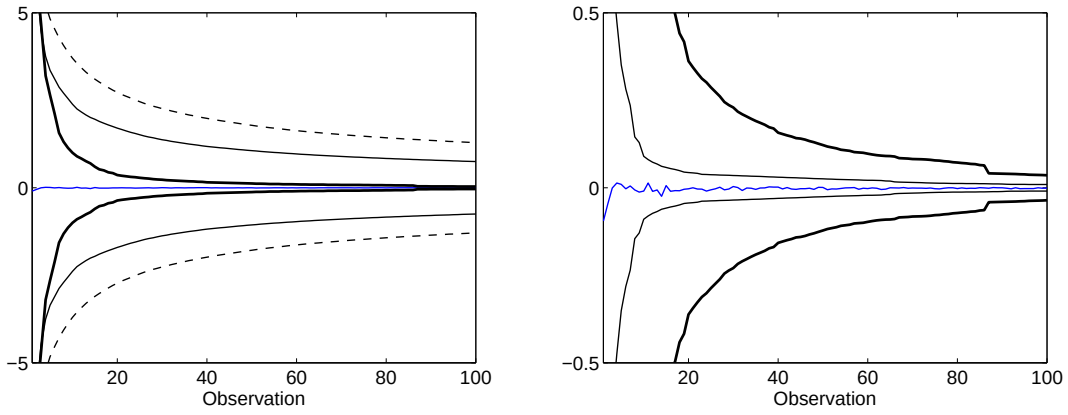
Fig. 3: Mean square estimation error based on 100 runs of the various algorithms. Left panel (a): showing the CSKF-1 (solid line), the CSKF-0.5 (marked by circles), the CSKF with the approximate l_0 norm (dashed line), the S^3 algorithm (thick solid line), and a regular KF that is aware of the signal support (dotted line). Right panel (b): showing the EBW when negative elements of z are allowed (dashed line), the EBW when all elements of z are non-negative (solid line), the CSKF-1 (thick line), and an ordinary KF that is aware of the actual support. Static case.

accuracy at each local CS step. At this point we have set $a = 0.05$ (which seemed to be fairly small for maintaining the local RIP at a sufficient level).

We have implement a CS-embedded EKF (or in short CS-EKF), which is essentially an EKF with a CS pseudo-measurement stage (see Section IV-A), for recovering z using the sequence of noisy observations $y_1, \ldots, y_k$. It should be noted that unlike the KF, the EKF is a suboptimal estimator that relies on the validity of the linearization assumption of small estimation errors. As such, it usually requires some tuning procedures to be carried out, e.g., the incorporation of artificial process noise. In this example we have set the process noise covariance as $Q = 5 \times 10^{-2} I_{256\times 256}$.

The estimation performance based on 100 Monte Carlo runs of the EKF variants (i.e., with either the l_p, $p = 1, 0.5$ norms or the approximate l_0 norm) is shown in Fig. 5. For comparison we have depicted the performance of two ordinary EKF's (i.e., without a local CS stage) that were implemented, one of which is aware of the actual support of the signal. As it can be clearly seen from the left panel in this figure, the local CS stage indeed improves the estimation performance over the ordinary EKF. Nevertheless, it seems that at least for this specific case, the various norm formulations of the CS stage yield roughly the same performance.

The same nonlinear problem was solved using the EBW. As before, the actual sparse signal is assumed to be normalized, i.e., $\| z \|_1 = 1$. The objective function used by the EBW is given



(a) **(b)**

Fig. 4: The estimation error $z(1) - \hat{z}(1)$ (middle line) and the 1σ bounds of: (a) the CSKF-1 (dashed line), the CSKF with the approximate l_0 norm (solid line), and the CSKF-0.5 (solid thick line). (b) the CSKF-0.5 with $N_\tau = 50$ PM iterations (thick line), and with $N_\tau = 200$ PM iterations. Static case.

by

$$G(z) = p(y \mid z) \propto \prod_{i=1}^{k} \exp\left\{ -\frac{1}{2} \left(y_i - h(z)\right)^T R_i^{-1} \left(y_i - h(z)\right) \right\} \tag{42}$$

where R_i denotes the observation noise covariance. The results of this experiment are shown in Fig. 5b. Surprisingly, the EBW outperforms all other methods while exhibiting a rapid convergence towards the EKF that is aware of the signal support. This superiority over the CS-EKF can be related to the guaranteed convergence of the EBW (in this case z is defined over a probability domain, see Theorem 1), a property that essentially depends on tuning and initial conditions in the case of the EKF.

D. Example: Lossy Speech Compression

In all previous examples the underlying signals were assumed to be sparse. The following case consider compressible signals which are not necessarily sparse. Analogously to the support in the sparse case, here the most significant elements in terms of magnitude comprises the set of interest.

Speech is a compressible signal. Usually, vowels can be represented using a limited number of frequencies for which the human hear is most sensitive. The cardinality of this set of significant frequencies may serve as an analog measure to sparseness degree $\#\{\mathrm{supp}(z)\}$. A more formal argument proceeds as follows. Let $z \in \mathbb{R}^n$ be the discrete Fourier transform (DFT) of y_k over

the discrete times $k = 1, \ldots, n$, that is

$$z(j) = \sqrt{n}^{-1} \sum_{k=1}^{n} y_k \exp\left(-\frac{2\pi i}{n}(j-1)(k-1)\right), \quad j = 1, \ldots, n \tag{43}$$

which can be compactly written as

$$z = \mathcal{F} y \tag{44}$$

where $\mathcal{F}$ and $y \in \mathbb{R}^n$ denote the DFT matrix and a vector whose components are the time points y_j, respectively. Denote F_ϵ the set of ϵ-significant frequencies, and let

$$F_\epsilon = \{z(j) \mid \quad 10 \log |z(j)| > \epsilon\} \tag{45}$$

that is, all frequencies for which the amplitude is greater than ϵ dB. Following this definition, $\#F_\epsilon$ is an analog measure to sparseness degree where $n/\#F_\epsilon$ is the compression ratio.

In this example we have used the CSKF for reconstructing a frequency representation z of a speech signal from under-sampled time series. In other words, our reconstruction algorithm solves the following problem

$$y = \mathcal{F}_m^* z + \zeta, \qquad y \in \mathbb{R}^m, \quad m < n \tag{46}$$

where $\mathcal{F}_m^* \in \mathbb{R}^{m \times n}$ denotes a sub-matrix obtained by sampling m rows from the inverse DFT matrix (which, in this case, is the conjugate transpose of $\mathcal{F}$). If we follow the arguments presented in [2] (Theorem 2.1) for sparse signals, we may say that in this case an adequate frequency representation is highly probable provided that

$$m \geq c \cdot \#F_\epsilon \log n \tag{47}$$

1) Experimental Settings: The CSKF was implemented using the approximate l_0 norm for reconstructing the short time DFT of a speech recording from a series of overlapping Hamming windows. The algorithm utilized $N_\tau = 200$ PM iterations with $R_\epsilon = 100^2$ and $\alpha = 1$. The window size was set to 256 with only 6 non-overlapping elements. In this example, our DFT vector z is composed out of $n = 256$ elements corresponding to the amplitude and phase of 128 frequencies. Taking the frequency threshold parameter $\epsilon = 0$ in (45) yields $\#F_\epsilon$ between 10 to 20 for the specific signal considered. A rough estimate based on (47) suggests that we need around $m = 110c$ samples picked at each time window for a 'good' frequency representation. We have tested the algorithm with both $m = 165$ and $m = 205$ samples, i.e., the algorithm uses either 65% or 80% of the available data. The results of these experiments are summarized in Figs. 6 and 7.

2) Results: The entire time series is shown in Fig. 6d. A typical random sampling pattern when using 65% of the samples in a single time window is shown in Fig. 6c. The original short time DFT of the signal (i.e., when using all available data) is depicted via a spectrogram in Fig. 6b. The reconstructed short time DFT based on the under-sampled data is shown in Fig. 6a. In a companion figure, Fig. 7, the performance of the algorithm is compared when using either 65% or 80% of the available data. The DFT reconstructions in both cases are shown in Figs. 7a

and 7b. These spectrograms are accompanied by slices at a single time point of the original and reconstructed signals. In this figures the original signal is shown via a dotted line. As it could be expected, the algorithm better captures the amplitudes of subtle frequencies when using 80% of the available data.

VIII. Conclusions

New methods are presented for sparse signal recovery from a sequence of noisy observations: 1) CSKF, 2) CS-EKF, 3) EBW, and 4) stochastic subset search (S^3). The CSKF, a Kalman filtering-based algorithm that was initially derived in [19]–[21], relies on a simple modification of the basic KF scheme. Three variants of this algorithm utilizing different norm relaxations are tested and compared in both static and dynamic scenarios. In all examples it is evident that the non-convex l_p, $0 < p < 1$ relaxation (CSKF-p) as well as the approximate l_0 norm improve the estimation accuracy with respect to the l_1 norm (CSKF-1).

The CS-EKF algorithm demonstrates the application of CS in the nonlinear case. It relies on the notion of local CS introduced inhere. The essential idea, which is exemplified by comparing the performance of the CS-EKF with an ordinary EKF, is that CS can be used for improving estimation accuracy in cases where the nonlinear sensing function maps the state onto a lower-dimensional observation space, $h(z) : \mathbb{R}^n \to \mathbb{R}^m$, $m < n$. Thus, local CS applies for sufficiently small and sparse perturbations from a nominal state.

The EBW, a popular optimization technique used in speech recognition, is amended here for directional sparse optimization (i.e., assuming a normalized signal). This method exhibits inferior estimation performance compared to the CSKF in the linear case. Its real advantage, however, is clear in the nonlinear case in which it outperforms the CS-EKF owing to its guaranteed convergence (proved inhere).

The last method derived in this work, the S^3, is based on a Markov chain Monte Carlo mechanism. It uses random finite set representations for modeling sparseness. In the simulations, the S^3 is shown to attain the highest estimation accuracy among all other methods. Its estimation performance, however, depends on the size of the particles population which makes it, in general, highly computationally demanding.

Appendix

A. Proof of Theorem 1

The following lemma is needed for proving the theorem.

Lemma 2: Let $F(z) = \tilde{F}(\{u_j\}) = \tilde{F}(\{g_j(z)\}) = \tilde{F} \circ g(z)$ where $u_j = g_j(z), j = 1,..m$ and z varies in some real vector space $\mathbb{R}^n$ of dimension n. Let $g_j(z)$ for all $j = 1,...m$ and $F(z)$ be differentiable at z. Assume that $\frac{\partial \tilde{F}(\{u_j\})}{\partial u_j}$ exists at $u_j = g_j(z)$ for all $j = 1,...m$. Let $L(z\prime) \equiv \nabla \tilde{F}\Big|_{g(z)} \cdot g(z\prime)$, $z\prime \in \mathbb{R}^n$. Let T_D be a family of transformations $\mathbb{R}^n \to \mathbb{R}^n$ such that for some $l = (l_1...l_n) \in \mathbb{R}^n$, $T_D(z) - z = l/D + o(1/D)$ if $D \to \infty$ (here $o(\epsilon)$ stands for the small 'o' notation, i.e., $o(\epsilon)/\epsilon \to 0$ for $\epsilon \to 0$). Assume that $T_D(z) = z$ if

$$\nabla L|_z \cdot l = 0 \tag{48}$$

Then for sufficiently large D, T_D is a growth for F at z iff T_D is a growth for L at z.

Proof of Lemma First, from the definition of L we have

$$\frac{\partial F(z)}{\partial z_k} = \sum_j \frac{\partial \tilde{F}(\{u_j\})}{\partial u_j}\frac{\partial g_j(z)}{\partial z_k} = \frac{\partial L(z)}{\partial z_k}$$

Next, for $z\prime = T_D(z)$ and sufficiently large D we have

$$F(z\prime) - F(z) = \sum_i \frac{\partial F(z)}{\partial z_i}(z_i\prime - z_i) + o(1/D) = \sum_i \frac{\partial F(z)}{\partial z_i} l_i/D + o(1/D)$$

$$= \sum_i \frac{\partial L(z)}{\partial z_i} l_i/D + o(1/D) = \sum_i \frac{\partial L(z)}{\partial z_i}(z_i\prime - z_i) + o(1/D) = L(z\prime) - L(z) + o(1/D)$$

Therefore for sufficiently large D, $F(z\prime) - F(z) > 0$ iff $L(z\prime) - L(z) > 0$.

Proof of Theorem Following the linearization principle, we first assume that $F(z) = l(z) = \sum a_{ij} z_{ij}$ is a linear form. Than the transformation formula for $l(x)$ is given by

$$\hat{z}_{ij} = \frac{a_{ij} z_{ij} + D z_{ij}}{l(z) + D} \tag{49}$$

We need to show that $l(\hat{z}) \geq l(z)$. It is sufficient to prove this inequality for each linear sub component associated with i, $\sum_{j=1}^{j=n} a_{ij}\hat{z}_{ij} \geq \sum_{j=1}^{j=n} a_{ij} z_{ij}$. Therefore without loss of generality we can assume that i is fixed and drop subscript i in the ensuing (i.e. we assume that $l(z) = \sum a_j z_j$, where $z = \{z_j\}$, $z_j \geq 0$ and $\sum z_j = 1$). We have: $l(\hat{z}) = \frac{l_2(z) + C l(z)}{l(z) + C}$, where $l_2(z) := \sum_j a_j^2 z_j$. The linear form of Theorem 1 follows in the next two lemmas.

Lemma 3:

$$l_2(z) \geq l(z)^2 \tag{50}$$

Proof of Lemma Let as assume that $a_j \geq a_{j+1}$. Substituting $z\prime = \sum_{j=1}^{j=n-1} z_j$ we need to show that

$$\sum_{j=1}^{j=n-1} [a_j^2 z_j + a_n^2(1 - z\prime)] \geq \sum_{j=1}^{j=n-1} (a_j - a_n)^2 z_j^2 + 2\sum_{j=1}^{j=n-1} (a_j - a_n) a_n z_j + a_n^2$$

We will prove the above relation by showing for every fixed j that $(a_j^2 - a_n^2) z_j \geq (a_j - a_n)^2 z_j^2 + 2(a_j - a_n) a_n z_j$. If $(a_j - a_n) z_j \neq 0$ then the above inequality is equivalent to $a_j - a_n \geq (a_j - a_n) z_j$ which obviously holds since $0 \leq z_j \leq 1$.

Lemma 4: For sufficiently large $|D|$ the following holds: $l(\hat{z}) > l(z)$ if D is positive and $l(\hat{z}) < l(z)$ if D is negative.

Proof of Lemma From (50) we have the following inequalities.

$$l_2(z) + D l(z) \geq l(z)^2 + D l(z)$$

$$l(\hat{z}) = \frac{l_2(z) + D l(z)}{l(z) + D} \geq \frac{l(z)^2 + D l(z)}{l(z) + D} \qquad \text{if} \quad l(z) + D > 0$$

$$l(\hat{z}) = \frac{l_2(z) + Dl(z)}{l(z) + D} \leq \frac{l(z)^2 + Dl(z)}{l(z) + D} \qquad \text{if} \quad l(z) + D < 0$$

Now, Theorem 1 follows immediately upon recognizing that (48) is equivalent to $l_2(z) - l(z)^2 = 0$ for large D.

REFERENCES

[1] E. J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information", *IEEE Transactions on Information Theory*, vol. 52, pp. 489–509, 2006.

[2] E. J. Candes, "Compressive sampling", Madrid, Spain, 2006, European Mathematical Society, Proceedings of the International Congress of Mathematicians.

[3] D. Donoho, "Compressed sensing", *IEEE Transactions on Information Theory*, vol. 52, pp. 1289–1306, 2006.

[4] M. Lustig, D. Donoho, and J. M. Pauly, "Sparse mri: The application of compressed sensing for rapid mr imaging", *Magnetic Resonance in Medicine*, vol. 58, pp. 1182–1195, 2007.

[5] U. Gamper, P. Boesiger, and S. Kozerke, "Compressed sensing in dynamic mri", *Magnetic Resonance in Medicine*, vol. 59, pp. 365–373, 2008.

[6] R. Chartrand, "Exact reconstruction of sparse signals via nonconvex minimization", *IEEE Signal Processing Letters*, vol. 14, pp. 707–710, 2007.

[7] R. Tibshirani, "Regression shrinkage and selection via the lasso", *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[8] E. Candes and T. Tao, "The dantzig selector: statistical estimation when p is much larger than n", *Annals of Statistics*, vol. 35, pp. 2313–2351, 2007.

[9] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit", *SIAM Journal of Scientific Computing*, vol. 20, no. 1, pp. 33 – 61, 1998.

[10] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression", *The Annals of Statistics*, vol. 32, no. 2, pp. 407 – 499, 2004.

[11] M. E. Tipping, "Sparse bayesian learning and the relevance vector machine", *Journal of Machine Learning Research*, vol. 1, pp. 211 – 244, 2001.

[12] R. E. McCulloch and E. I. George, "Approaches for bayesian variable selection", *Statistica Sinica*, vol. 7, pp. 339 – 374, 1997.

[13] J. Geweke, *Bayesian Statistics 5*, chapter Variable selection and model comparison in regression, Oxford University Press, 1996.

[14] B. A. Olshausen and K. Millman, "Learning sparse codes with a mixtureof-gaussians prior", *Advances in Neural Information Processing Systems (NIPS)*, pp. 841 – 847, 2000.

[15] S. J. Godsil and P. j. Wolfe, "Bayesian modelling of time-frequency coefficients for audio signal enhancement", *Advances in Neural Information Processing Systems (NIPS)*, 2003.

[16] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries", *IEEE Transactions on Signal Processing*, vol. 4, pp. 3397 – 3415, 1993.

[17] Y. C. Pati, R. Rezifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition", 27th Asilomar Conf. on Signals, Systems and Comput., 1993.

[18] S. Chen, S. A. Billings, and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification", *International Journal of Control*, vol. 50, pp. 1873 – 1896, 1989.

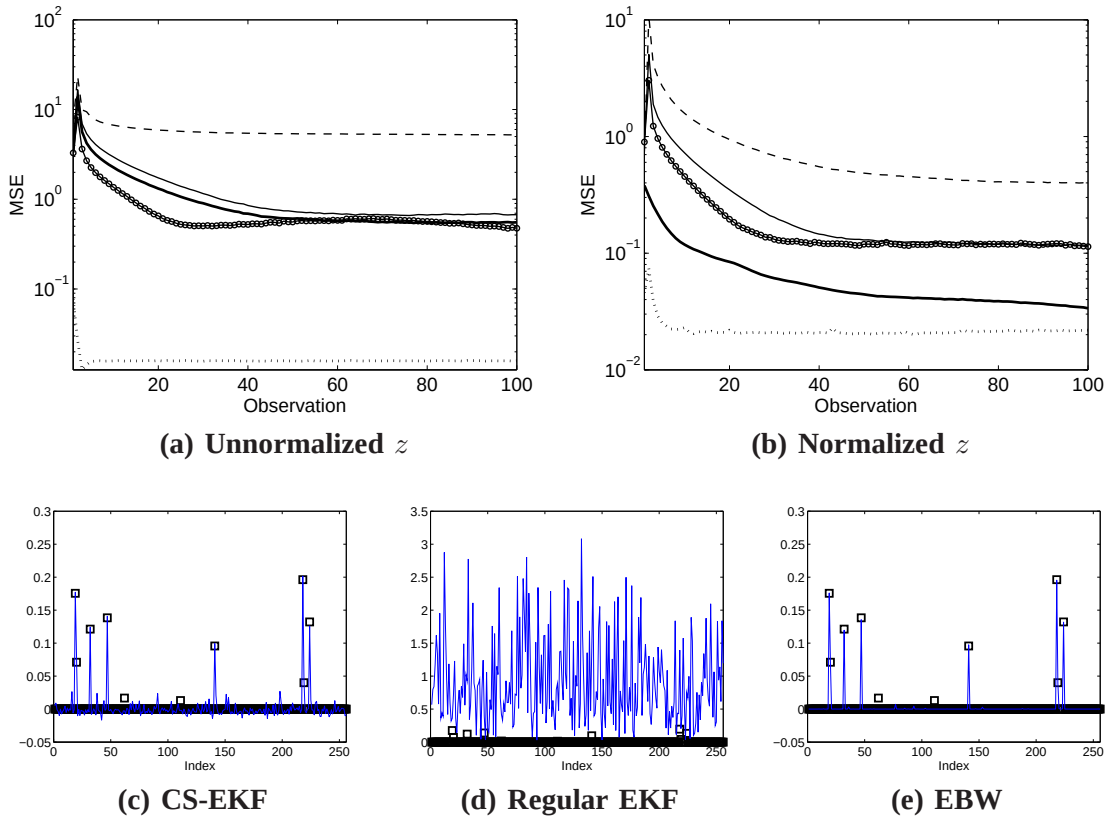
[19] A. Carmi, P. Gurfil, and D. Kanevsky, "A simple method for sparse signal recovery from noisy observations using kalman filtering", Tech. Rep. RC24709, Human Language Technologies, IBM, 2008.

[20] A. Carmi, P. Gurfil, and D. Kanevsky, "A simple method for sparse signal recovery from noisy observations using kalman filtering. embedding approximate quasi-norm for improved accuracy", Tech. Rep. RC24711, Human Language Technologies, IBM, 2008.

[21] A. Carmi, P. Gurfil, and D. Kanevsky, "Methods for sparse signal recovery using kalman filtering with embedded pseudo-measurement norms and quasi-norms", *Submitted to IEEE Transactions on Signal Processing*, 2009.

[22] N. Vaswani, "Kalman filtered compressed sensing", October 2008, Proceedings of the IEEE International Conference on Image Processing (ICIP).

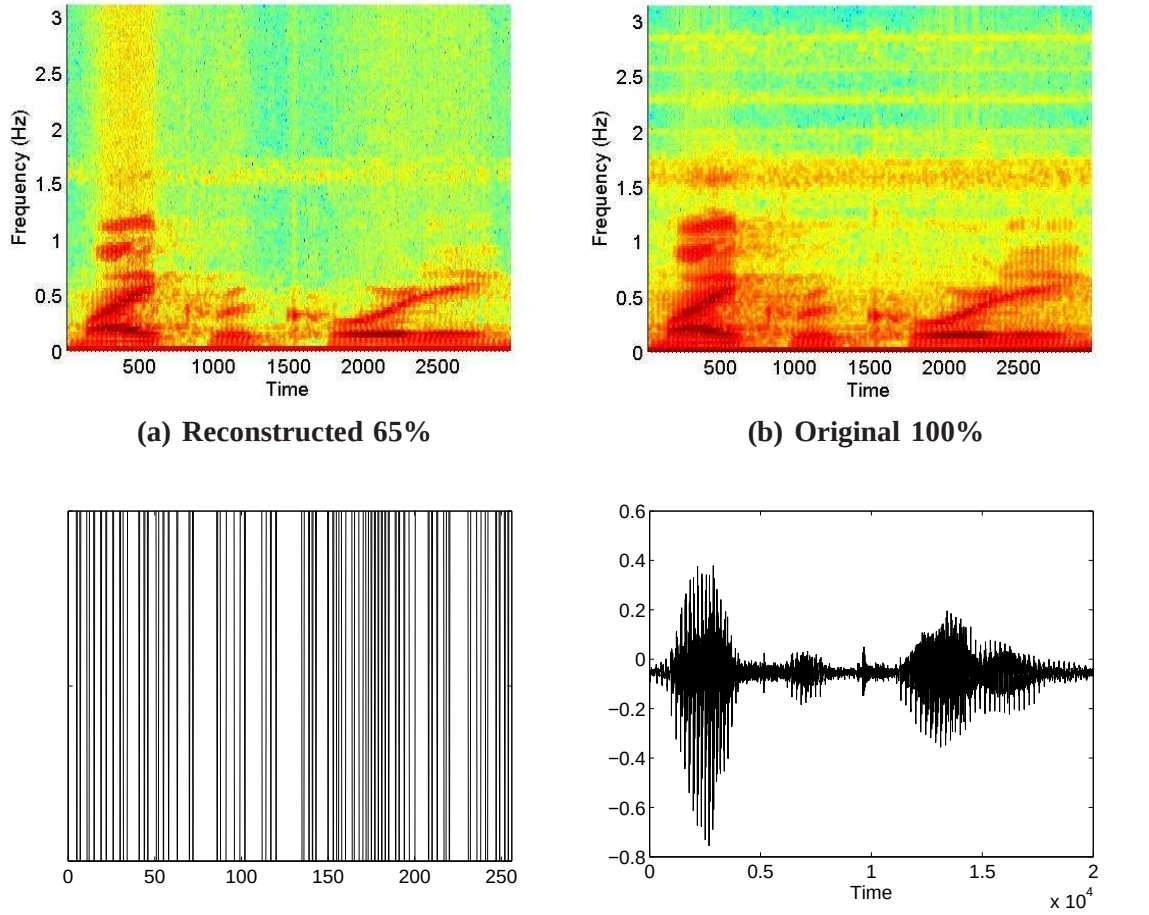
[23] J. M. Mendel, *Lessons in Estimation Theory for Signal Processing, Communications, and Control*, Prentice Hall, 1995.
[24] S. J. Julier and J. J. LaViola, "On kalman filtering with nonlinear equality constraints", *IEEE Transactions on Signal Processing*, vol. 55, no. 6, pp. 2774 – 2784, 2007.
[25] J. Deurschmann, I. Bar-Itzhack, and G. Ken, "Quaternion normalization in spacecraft attitude determination", American Institute of Aeronautics and Astronautics, 1992, pp. 27–37, Proceedings of the AIAA/AAS Astrodynamics Conference.
[26] T. Blumensath and M. E. Davies, "Gradient pursuits", *IEEE Transactions on Signal Processing*, vol. 56, no. 6, pp. 2370 – 2382, 2008.
[27] W. Johnson and J. Lindenstrauss, "Extensions of lipschitz maps into a hilbert space", *Contemporary Mathematics*, vol. 26, pp. 189 – 206, 1984.
[28] P.S. Gopalakrishnan, D. Kanevsky, D. Nahamoo, and A. Nadas, "An Inequality for Rational Functions with Applications to Some Statistical Estimation Problems", *IEEE Trans. Information Theory*, vol. 37, no. 1, January 1991.
[29] L.E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains", *Ann. Math. Statist.*, vol. 41, no. 1, pp. 164–171, 1970.
[30] D. Kanevsky, "Extended Baum Transformations for General Functions", in *Proc. ICASSP*, 2004.
[31] T. N. Sainath, D. Kanevsky, and B. Ramabhadran, "Gradient Steepness Metrics Using Extended Baum-Welch Transformations for Universal Pattern Recognition Tasks", in *Proc. ICASSP*, April 2008.
[32] D. Povey, *Discriminative Training for Large Vocabulary Speech Recognition*, PhD thesis, Cambridge University, 2003.
[33] P. J. Green, "Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination", *Biometrika*, vol. 82, no. 4, pp. 711–732, December 1995.
[34] A. Doucet, J. F. G. de Freitas, and N. J. Gordon, *Sequential Monte Carlo Methods in Practice*, New York: Springer-Verlag, 2001.
[35] O. Cappé, S.J. Godsill, and E.Moulines, "An overview of existing methods and recent advances in sequential monte carlo", *Proc. IEEE*, vol. 95, no. 5, pp. 899–924, May 2007.
[36] S. J. Godsill, "On the relationship between markov chain monte carlo methods for model uncertainty", *J. Comp. Graph. Stats*, vol. 10, no. 2, pp. 230–248, 2001.
[37] S. Chib and E. Greenberg, "Understanding the Metropolis-Hastings Algorithm", *The American Statistician*, vol. 49, no. 4, pp. 327–335, November 1995.
[38] S. K. Pang, J. Li, and S. J. Godsill, "Models and Algorithms for Detection and Tracking of Coordinated Groups", Proceedings of the IEEE 5th International Symposium on Image and Signal Processing Analysis, September 2007, pp. 504–509.



(a) Unnormalized z **(b) Normalized z**

(c) CS-EKF **(d) Regular EKF** **(e) EBW**

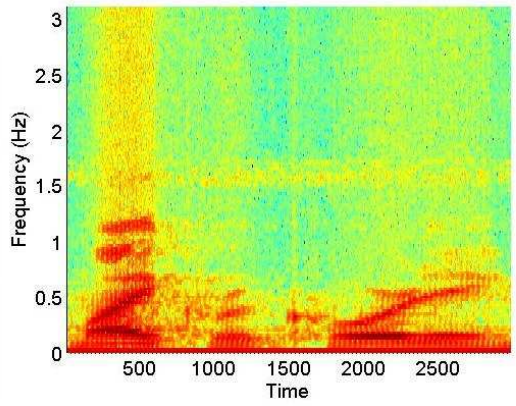
Fig. 5: Top panel: Mean square estimation error based on 100 Monte Carlo runs. (a) Showing the CS-EKF with l_1 norm (marked by circles), the CS-EKF with the approximate l_0 norm (thick line), the CS-EKF with l_p, $p = 0.5$ norm (solid line), and an ordinary EKF that is unaware (dashed line) and aware (dotted line) of the actual support. (b) Showing the CS-EKF with l_1 norm (marked by circles), the CS-EKF with the approximate l_0 norm (solid line), the EBW (thick line), and an ordinary EKF that is unaware (dashed line) and aware (dotted line) of the actual support. Bottom panel: Snapshot at $k = 100$ of the true (squares) and estimated (lines) signals using the CS-EKF (c), a regular EKF (d), and the EBW (e). Nonlinear case.



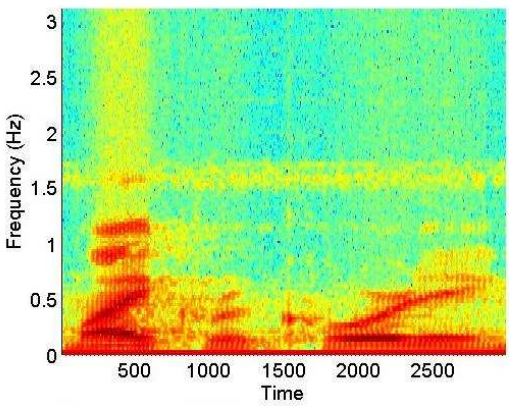
(a) Reconstructed 65% **(b) Original 100%**

(c) Sampling window **(d) Time domain**

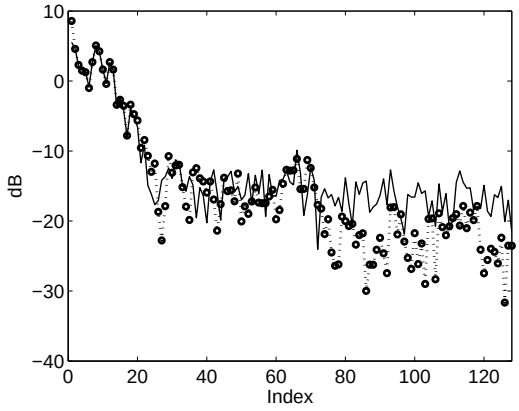
Fig. 6: Reconstructing a short time DFT of a speech signal from under-sampled data.



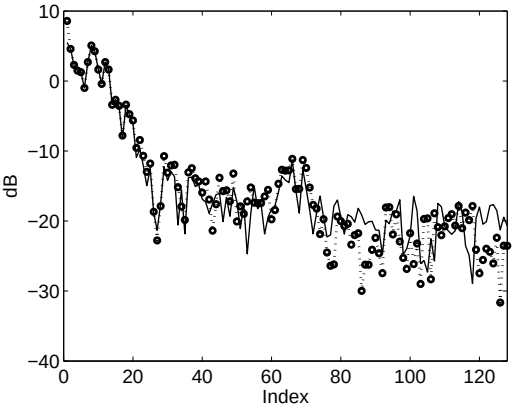
(a) Reconstructed 65%



(b) Reconstructed 80%



(c) Amplitude of Fourier coefficients 65%



(d) Amplitude of Fourier coefficients 80%

Fig. 7: Reconstructing a short time DFT of a speech signal from under-sampled data.