Tracking Dynamic Sparse Signals with Hierarchical Kalman Filters: A Case Study

Jason Filos, Evripidis Karseras and Wei Dai
Electrical and Electronic Engineering
Imperial College London
London, UK
{j.filos, e.karseras11, wei.dai1}@imperial.ac.uk

Shulin Yan
Department of Computing
Imperial College London
London, UK
shu.yan09@imperial.ac.uk

***Abstract*—Tracking and recovering dynamic sparse signals using traditional Kalman filtering techniques tend to fail. Compressive sensing (CS) addresses the problem of reconstructing signals for which the support is assumed to be sparse but is not fit for dynamic models. This paper provides a study on the performance of a hierarchical Bayesian Kalman (HB-Kalman) filter that succeeds in promoting sparsity and accurately tracks time varying sparse signals. Two case studies using real-world data show how the proposed method outperforms the traditional Kalman filter when tracking dynamic sparse signals. It is shown that the Bayesian Subspace Pursuit (BSP) algorithm, that is at the core of the HB-Kalman method, achieves better performance than previously proposed greedy methods.**

***Index Terms*—Kalman filtering; compressed sensing; sparse Bayesian learning; sparse representations.**

I. Introduction

In this work we consider the problem of reconstructing time sequences of signals that are assumed to be sparse in some transform domain. Recent studies have shown that sparse signals can be recovered accurately using less observations than what is considered necessary using traditional sampling criteria using the theory of compressed sensing (CS) [1], [2]. However, there are a number of practical limitations. First, the recovery of sparse signals using CS consists of solving an NP-hard minimization problem [1], [3]. Secondly, CS reconstruction is not fit for dynamic models. Existing solutions that address the dynamic problem either treat the entire time sequence as a single spatiotemporal signal and perform CS to reconstruct it [4], or alternatively apply CS at each time instance separately [5]. In [6] a non-Bayesian CS reconstruction is presented that assumes known sparsity levels and a non-dynamic model. A class of adaptive filters, based on the least mean squares (LMS) algorithm, are presented in [7] and [8]. The adaptive framework of the LMS is used in these approaches for CS. However, these approaches are not fit for dynamic sparse signal reconstruction.

For a single time instance, the problem of sparse signal reconstruction was solved in [9] using a Bayesian network that elegantly promotes sparsity. This learning framework, referred to as the Relevance Vector Machine (RVM), results in highly sparse models for the input and has gained popularity in the signal processing community for its use in compressed sensing applications [5] and basis selection [10]. Sparsity is rendered possible with a hierarchy of prior distributions that intuitively confines the space of all possible states. The key fact behind this technique is that it provides estimates on full distributions. Non-Bayesian sparse recovery algorithms do not take into account the signals' statistics making their use in tracking sparse signals difficult. The resulting statistical information can be used to make predictions for future states without the risk of them not being sparse.

For multiple time instances the system state can be tracked accurately using Kalman filtering. Unfortunately the classic Kalman approach is not fit for sparse signals. A Kalman filtering-based CS approach was first presented in [11] where the filter is externally modified to admit sparse solutions. The idea in [11] and [12] is to enforce sparsity by applying threshold operators. Work in [13] adopts a probabilistic model but signal amplitudes and support are estimated separately. Finally, the techniques presented in [14] use prior sparsity knowledge into the tracking process. All these approaches typically require a number of parameters to be pre-set.

In this work a sparsity-promoting Bayesian network is employed that extends the data model adopted in traditional tracking. The problem of sparse signal recovery is tackled efficiently by the hierarchical nature of the Bayesian model. The statistical information that is obtained as a by-product of the hierarchical model is then incorporated in updating previous estimates to produce sparsity-aware state estimates. Additionally, the automatic determination of the active components solves the problem of having to assume fixed sparsity levels and involves only the noise parameter as the unknown parameter to be adjusted manually. The hierarchical Bayesian Kalman filter (Section II) proposed in [15], [16], is tested on real data: First, the time-domain signal of a real piano recording is reconstructed using only 25% of originally sampled data (Section III). Second, incomplete satellite data, measuring the distribution of ozone gas in the planets' atmosphere, is recovered to yield the original content (Section IV). On basis of these two case studies, simulations show how the proposed method outperforms both the traditional Kalman filter, when dealing with dynamic sparse signals, and a state-of-the-art greedy CS reconstruction algorithm.

II. System Model

Let the random variables $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$, at some time t, describe the system state and observations respectively. The standard Kalman filter formulation is given by

$$\boldsymbol{x}_t = \boldsymbol{x}_{t-1} + \boldsymbol{q}_t, \tag{1}$$

$$\boldsymbol{y}_t = \boldsymbol{\Phi}_t \boldsymbol{x}_t + \boldsymbol{n}_t, \tag{2}$$

where $\boldsymbol{q}_t$ and $\boldsymbol{n}_t$ denote the process and measurement noise and $\boldsymbol{\Phi}_t$ is a design matrix. We assume that the signal $\boldsymbol{x}_t \in \mathbb{R}^n$ is sparse in some transform domain and that the entries $\boldsymbol{\Phi}_t \in \mathbb{R}^{m \times n}$ are independently identically distributed Gaussian random variables drawn from $\mathcal{N}\left(0, \frac{1}{m}\right)$.

Based on the Gaussian assumption one has $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_{t-1}, \boldsymbol{Q}_t)$ and $p\left(\boldsymbol{y}_t|\boldsymbol{x}_t\right) = \mathcal{N}\left(\boldsymbol{\Phi}\boldsymbol{x}_t, \sigma^2\boldsymbol{I}\right)$, with $p(\boldsymbol{q}_t) = \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{Q}_t\right)$ and $p(\boldsymbol{n}_t) = \mathcal{N}\left(\boldsymbol{0}, \sigma^2\boldsymbol{I}\right)$, so that the Kalman filter continuously alternates between the prediction and update step. The prediction step calculates the parameters of $p(\boldsymbol{x}_t|\boldsymbol{y}_{t-1})$ while the update step evaluates those of $p(\boldsymbol{x}_t|\boldsymbol{y}_t)$.

In order to address the sparsity of the state vector, an additional level of parameters $\boldsymbol{\alpha}$ is introduced to control the variance of each component x_i [9]:

$$p\left(\boldsymbol{x}|\boldsymbol{\alpha}\right) = \prod_{i=1}^{n} \mathcal{N}\left(0, \alpha_i^{-1}\right) = \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{A}^{-1}\right),$$

where matrix $\boldsymbol{A} = \mathrm{diag}\left([\alpha_1, \cdots, \alpha_n]\right)$. By driving $\alpha_i = +\infty$ it follows that $p\left(x_i|\alpha_i\right) = \mathcal{N}\left(0, 0\right)$ and consequently it is certain that $x_i = 0$.

A. Hierarchical Kalman filter

The principles behind the Kalman filter and SBL are put together to derive the Hierarchical Bayesian Kalman (HB-Kalman) filter. The proposed approach has several advantages. First, by adopting the same system model as described in Equations (1) and (2), one can track the mean and covariance of the state vector $\boldsymbol{x}_t$. Second, the employment of hyper-parameters, to model state innovation, promotes sparsity.

The measurement noise is chosen to be Gaussian with known covariance, i.e. $\boldsymbol{n} \sim \mathcal{N}\left(0, \sigma^2\boldsymbol{I}\right)$. The state innovation process is given by $\boldsymbol{q}_t \sim \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{A}_t^{-1}\right)$, with $\boldsymbol{A}_t = \mathrm{diag}\left(\boldsymbol{\alpha}_t\right) = \mathrm{diag}\left([\alpha_1, \cdots, \alpha_n]_t\right)$ and the hyper-parameters α_i are learned *online* from the data, as opposed to the traditional Kalman filter where the covariance matrix $\boldsymbol{Q}$ of $\boldsymbol{q}_t$ is given.

Similar to the standard Kalman filter, two steps, prediction and update, need to be performed at each time instance. In the prediction step, one has to evaluate:

$$\begin{aligned} &\boldsymbol{x}_{t|t-1} = \boldsymbol{x}_{t-1},\ \boldsymbol{\Sigma}_{t|t-1} = \boldsymbol{\Sigma}_{t-1} + \boldsymbol{A}_t^{-1}, \\ &\boldsymbol{y}_{t|t-1} = \boldsymbol{\Phi}_t \boldsymbol{x}_{t|t-1},\ \boldsymbol{y}_{e,t} = \boldsymbol{y}_t - \boldsymbol{y}_{t|t-1}. \end{aligned} \tag{3}$$

where the notation $t|t-1$ means prediction at time instance t for measurements up to time instance $t-1$. In the update step, one computes:

$$\begin{aligned} \boldsymbol{x}_{t|t} = \boldsymbol{x}_{t|t-1} + \boldsymbol{K}_t\boldsymbol{y}_{e,t}, \boldsymbol{\Sigma}_{t|t} = (\boldsymbol{I} - \boldsymbol{K}_t\boldsymbol{\Phi}_t)\boldsymbol{\Sigma}_{t|t-1}, \\ \boldsymbol{K}_t = \boldsymbol{\Sigma}_{t|t-1}\boldsymbol{\Phi}_t^T(\sigma^2\boldsymbol{I} + \boldsymbol{\Phi}_t\boldsymbol{\Sigma}_{t|t-1}\boldsymbol{\Phi}_t^T)^{-1}. \end{aligned}$$

Differently from the standard Kalman filter, one has to perform the additional step of learning the hyper-parameters $\boldsymbol{\alpha}_t$. From Equation (3) we get $\boldsymbol{y}_{e,t} = \boldsymbol{\Phi}_t\boldsymbol{q}_t + \boldsymbol{n}_t$ where a sparse $\boldsymbol{q}_t$ is preferred to produce a sparse $\boldsymbol{x}_t$. Following the analysis in [9] and [17], maximising the likelihood $p(\boldsymbol{y}_t|\boldsymbol{\alpha}_t)$ is equivalent to minimising the following cost function:

$$\mathcal{L}(\boldsymbol{\alpha}_t) = \log|\boldsymbol{\Sigma}_\alpha| + \boldsymbol{y}_{e,t}^T\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}^{-1}\boldsymbol{y}_{e,t}, \tag{4}$$

where $\boldsymbol{\Sigma}_{\boldsymbol{\alpha}} = \sigma^2\boldsymbol{I} + \boldsymbol{\Phi}_t\boldsymbol{A}_t^{-1}\boldsymbol{\Phi}_t^T$.

B. Bayesian Subspace Pursuit

The algorithms described in [17], that optimise cost function (4), are greedy algorithms at heart. An important observation can be made by deriving the scaled version of this cost function with the noise variance. After some basic linear algebra manipulation and for a single time instance the function becomes:

$$\begin{aligned} \sigma^2\mathcal{L} = \sigma^2\log\left|\sigma^2\boldsymbol{I} + \boldsymbol{\Phi}_{\mathcal{I}}\boldsymbol{A}_{\mathcal{I}}^{-1}\boldsymbol{\Phi}_{\mathcal{I}}^T\right| + \\ \boldsymbol{y}^T\left(\boldsymbol{I} - \boldsymbol{\Phi}_{\mathcal{I}}(\sigma^2\boldsymbol{A}_{\mathcal{I}} + \boldsymbol{\Phi}_{\mathcal{I}}^T\boldsymbol{\Phi}_{\mathcal{I}})^{-1}\boldsymbol{\Phi}_{\mathcal{I}}^T\right)\boldsymbol{y}. \end{aligned} \tag{5}$$

Subscript $\mathcal{I}$ denotes the subset of columns of $\boldsymbol{\Phi}$ for which $0 < \alpha_i < +\infty$. By taking the limit of Equation (5) for when noise variance approaches zero we obtain:

$$\lim_{\sigma^2 \to 0} \sigma^2\mathcal{L}\left(\boldsymbol{\alpha}\right) = \left\|\boldsymbol{y} - \boldsymbol{\Phi}_{\mathcal{I}}\boldsymbol{\Phi}_{\mathcal{I}}^{\dagger}\boldsymbol{y}\right\|_2^2. \tag{6}$$

This result suggests that the principle behind the optimisation of the cost function is the same as the one used in Basis Pursuit and subsequently many greedy sparse reconstruction algorithms such as the OMP [18] and Subspace Pursuit [19]. In fact it can be shown that the proposed algorithm requires more strict bounds on the mutual coherence of design matrix $\boldsymbol{\Phi}$ than the OMP algorithm. The proposed algorithm described in Algorithm 1 adopts attributes from the Subspace Pursuit into the optimisation procedure. The qualities of the inference procedure are consequently improved. For a detailed analysis of the mathematical derivations the interested reader is referred to [16]. For clarity the fundamental quantities needed for Algorithm 1 are listed below. These are the scaled versions of the corresponding quantities derived in [17]:

$$\sigma^2 s_i = \boldsymbol{\phi}_i^T\left(\sigma^2\boldsymbol{C}_{-i}^{-1}\right)\boldsymbol{\phi}_i, \quad \sigma^2 q_i = \boldsymbol{\phi}_i^T\left(\sigma^2\boldsymbol{C}_{-i}^{-1}\right)\boldsymbol{y}.$$

The motivation for deriving scaled versions of the quantities given in [17] is the poor performance of the original derivations when the noise variance is known *a-priori*. It is simple to ascertain this fact by setting $\sigma^2 = 0$ in [17]. This is an important observation since in most real world tracking scenarios the noise floor is usually estimated or labelled by the sensor manufacturer.

III. Audio signal reconstruction

We consider the problem of reconstructing a recording of a classical piano piece in a real reverberant environment. The recorded signal is highly non-stationary, broadband and contains overlapping notes (that might be harmonically related). To make things worse, the pedal on the piano is engaged

Algorithm 1 Bayesian Subspace Pursuit

Input: $\mathbf{\Phi}, \boldsymbol{y}, \sigma^2$

Initialise:
- Initialise iteration counter, $k = 1$.
- Calculate $\theta_i = |\boldsymbol{\phi}_i^T \boldsymbol{y}| - 1$ for $i \in [1, n]$.
- $T_k = \{i \in [1, n] : \theta_i > 0\}$.
- If $|T_k| = 0$ then $T_k = \{$index $i \in [1, n]$ for which $|\boldsymbol{\phi}_i^T \boldsymbol{y}|$ is maximised$\}$.
- Calculate $\alpha_i = \frac{1}{|\boldsymbol{\phi}_i^T \boldsymbol{y}|}$ for $i \in T_k$.

Iteration:
- Store $\alpha_{max} = \arg\max_{i \in T_k} |\alpha_i|$.
- Calculate values α_i and $\theta_i = (\sigma^2 q_i)^2 - (\sigma^2 s_i)$ for $i \in [1, n]$.
- Construct subsets: $T_{\theta_i > 0} = \{i \in [1, n] : \theta_i > 0\}$ and $T_{\alpha_i \leq a_{max}} = \{i \in [1, n] : |\alpha_i| \leq a_{max}, |\alpha_i| < +\infty\}$.
- If $|T_{\theta_i > 0}| = 0$ then $s = |T_{\alpha_i \leq a_{max}}| + 1$ else $s = |T_{\theta_i > 0}| + |T_{\alpha_i \leq a_{max}}|$.
- $T' = T_k \cup \{$indices corresponding to s smallest values of α_i for $i \in [1, n]\}$.
- Increase counter, $k = k + 1$.
- Compute covariance matrix $\sigma^{-2}\mathbf{\Sigma}_x$ and $\boldsymbol{\mu}_x$.
- $T_k = \{$indices corresponding to s largest non-zero values of $|\boldsymbol{\mu}_x|$ for which $\alpha_i > 0\}$
- If $\sigma^2 \mathcal{L}_k$ has reached a steady state then quit. Continue otherwise.

Output:
- Estimated support set T_k and sparse signal $\hat{\boldsymbol{x}}$ with $|T_k|$ non-zero components, $\hat{\boldsymbol{x}}_T = \boldsymbol{\mu}_x$.
- Estimated covariance matrix $\sigma^{-2}\mathbf{\Sigma}_x$.

throughout, causing significant time-frequency smearing. The piano recording is sampled at a frequency of $f_s = 44.1$ kHz and split into T non-overlapping frames of length $n = 1024$ samples.

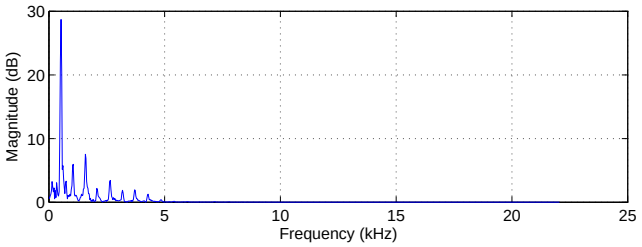


Fig. 1. Snapshot of the sparse frequency content in a single time-frame of piano data.

A. Experimental setup

The support of each frame in the frequency domain is assumed to be sparse i.e., there is only a small number of frequencies present, as can be seen from Figure 1. For each of these frames only a number of m samples are kept. The set $\mathcal{I}_t \subset [1, n]$ contains the time indices of the samples kept for each block and is chosen uniformly at random. In analogy with Equations (1) and (2), $\boldsymbol{y}_t \in \mathbb{R}^m$ denotes the random samples taken from each block at time instance t, $\boldsymbol{x}_t \in \mathbb{C}^n$ denotes the support of each frame in the frequency domain and matrix $\mathbf{\Phi}_t = \mathcal{F}_{\mathcal{I}_t}^{-1} \in \mathbb{C}^{m \times n}$ is formed by keeping the rows of matrix $\mathcal{F}^{-1}$ with indices in the set $\mathcal{I}_t$. Matrix $\mathcal{F} \in \mathbb{C}^{n \times n}$ denotes the Fourier basis matrix.

B. Results

At each time instance an estimate for the support $\boldsymbol{x}_t$ is recovered. Since the assumed basis is the Fourier basis the support is tracked in the both the real and imaginary domain. The measurement matrices now become $\Re\{\mathbf{\Phi}_t\}$ and $\Im\{\mathbf{\Phi}_t\}$. Note that these matrices belong in $\mathbb{R}^{m \times n/2}$ because of the symmetry of the Fourier transform for real-valued functions. The simulation time for this experiment is $T = 100$ time domain frames. At each time instance $m = 256$ samples are kept. Measurement noise variance is set to the sufficiently small value of $\sigma^2 = 0.1^5$ for the entire simulation time.

The root-mean-square error (RMSE) is evaluated at each time instance t such that

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} y_i - \hat{y}_i},$$

where y_i and $\hat{y}_i$ are the n individual elements of the recovered time-domain signal and the original signal respectively. Three

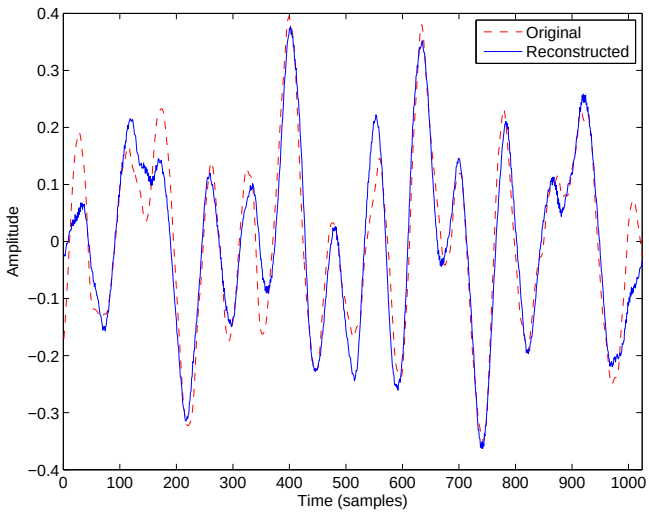


Fig. 2. Original time-domain representation (dotted line) of a frame of audio, along with the reconstructed data using the HB-Kalman.

recovery algorithms are considered, viz. standard Kalman filtering, BCS [5], and the proposed method [15], [16].

The resulting RMSE for the whole simulation time T is shown in Figure 3. As can be seen from the resulting graph, the error levels are much lower for the HB-Kalman filter when compared to the standard Kalman filter; this is a direct consequence of the assumed sparse model. By comparing to the repeated application of the BCS method, i.e assuming independent, identically distributed data, we see that incorporating statistical information from previous estimates results in lower reconstruction error. It is important to emphasise

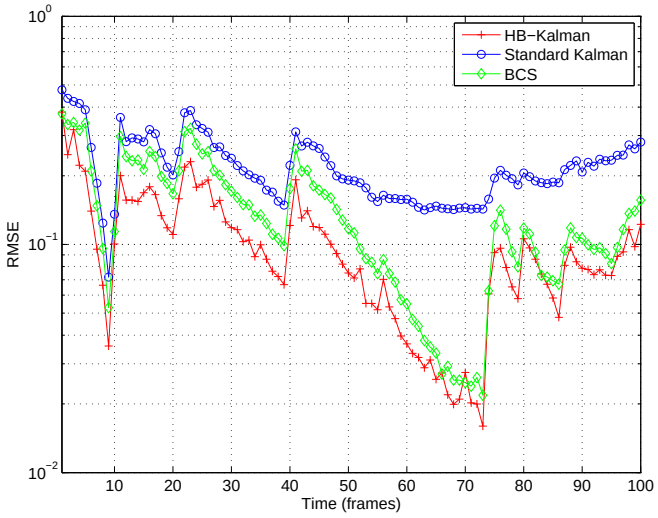


Fig. 3. Reconstruction error given over a time period of 100 frames. Each frame contains $n = 1024$ samples from which $m = 256$ are chosen at random.

that the parameters are identical for both the BCS algorithm and the HB-Kalman for a fair comparison; tuning certain parameters individually for each algorithm can lead to better reconstruction results depending on the specific application scenario.

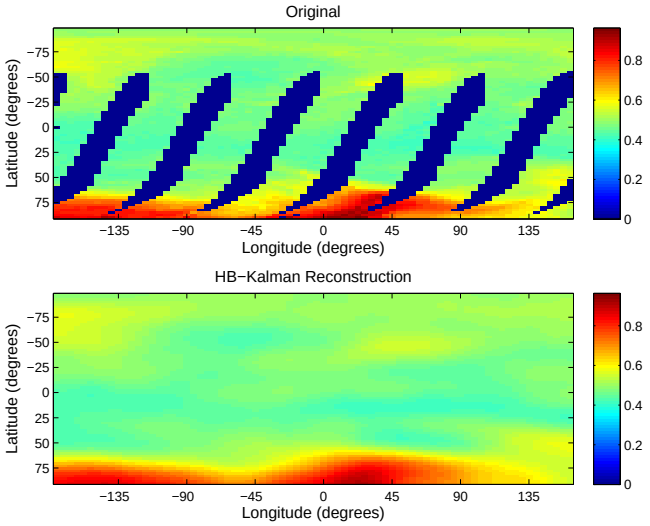
IV. Ozone distribution signal reconstruction

Here, we aim to reconstruct an ozone distribution signal from a limited number of samples. The satellite data for this experiment is obtained from the Ozone Monitoring Instrument (OMI) on the NASA Aura spacecraft [20], that provides near-real-time measurements of ozone in the atmosphere. The test data is taken during a one month period on January 2012. The blue vertical strips, that can be seen in the top half of Figures 4(a)–(b), represent missing ozone measurements. They are due to the trajectory of the satellite around the rotating earth.

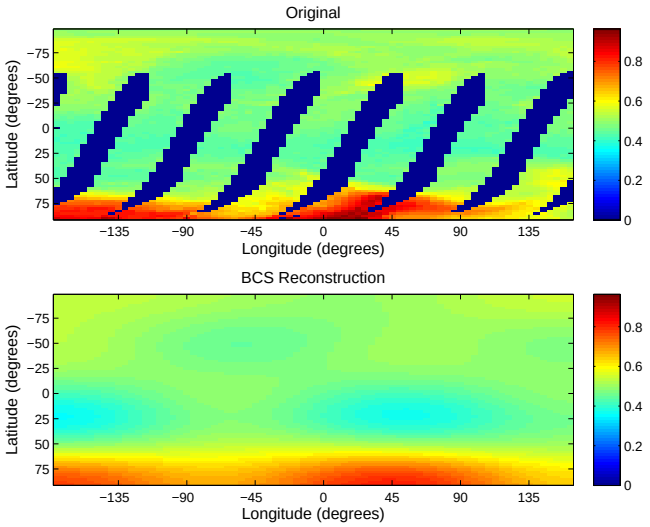
A. Experimental setup

The original data is cropped to a square image and undersampled by a factor of 8. The resolution of the resulting map is 75×75 corresponding to $n = 5625$. The area covered corresponds to Latitudes 90 degrees South to 90 degrees North and Longitudes 180 degrees West to 180 degrees East.

We assume that the support of each block in the discrete cosine transform (DCT) domain is sparse. For each block only a number of m samples are available, equal to approximately 75% of the observed data. The set $\mathcal{J}_t \subset [1, n]$ contains the indices of the samples that correspond to nonzero elements in the original *incomplete* data. Similar to the example in the previous Section, $\boldsymbol{x}_t \in \mathbb{C}^n$ denotes the support of each block in the DCT domain and matrix $\boldsymbol{\Phi}_t = \mathcal{G}_{\mathcal{J}_t}^{-1} \in \mathbb{C}^{m \times n}$ is formed by keeping the rows of matrix $\mathcal{G}^{-1}$ with indices in the set $\mathcal{J}_t$, where matrix $\mathcal{G} \in \mathbb{C}^{n \times n}$ denotes the DCT basis matrix.



(a) HB-Kalman



(b) BCS

Fig. 4. Atmospheric ozone distribution measured in (normalised) Dobson units. The original data, displayed on the top of each figure respectively, includes missing data. Reconstructed data using the HB-Kalman (a) and the BCS approach (b) displayed on the bottom respectively.

B. Results

At each day an estimate for the support $\boldsymbol{x}_t$ is recovered, as can be seen in Figures 4(a)–(b). The simulation time for this experiment is $T = 31$ days. At each time instance $m = 4275$ samples are observed from the data. Measurement noise variance is set to $\sigma^2 = 0.1^6$ for the entire simulation time.

The root-mean square error (RMSE) is evaluated at each time instance t such that

$$\text{RMSE} = \sqrt{\frac{1}{|\mathcal{J}_t|} \sum_{i \in \mathcal{J}_t} y_i - \hat{y}_i},$$

where y_i and $\hat{y}_i$ are the $|\mathcal{J}_t|$ individual pixels of the original

and recovered ozone map, respectively, represented as a vector. Note, that the error is calculated only for pixels that are present in both the original and the reconstructed image; missing data is ignored in the total RMSE since there are no ground-truths to compare the predicted data to.

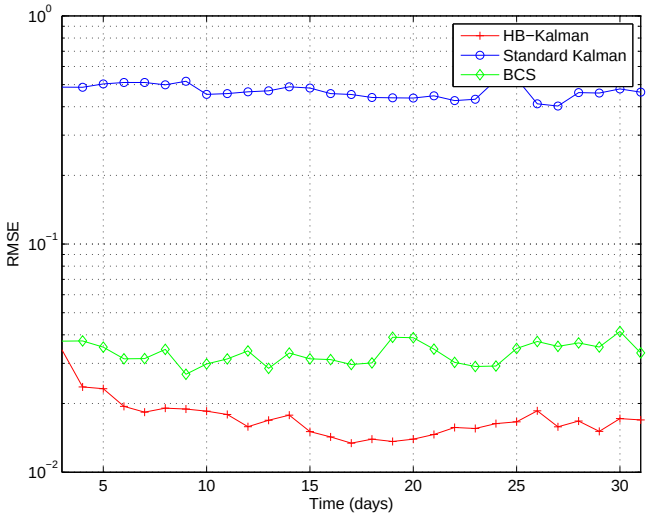


Fig. 5. Reconstruction error given over a time period of one month for the ozone distribution data.

The reconstruction accuracy is evaluated over the whole one month time period. As can be seen from Figure 5, the standard Kalman filter fails to accurately track the dynamic sparse signal. Additionally, although not explicit from this figure, the standard Kalman filter does not reconstruct the missing data. By contrast, repeated application of the BCS method and the HB-Kalman filter exhibit much lower error levels and accurately reconstruct the missing data. Furthermore, it can be seen that the HB-Kalman method outperforms the BCS method since it incorporates statistical information from previous days resulting in lower reconstruction error over the whole time period under analysis.

V. Conclusions

This paper presented a study on the performance of a novel hierarchical Bayesian Kalman filter, first proposed by the authors in [15], [16], that succeeds in promoting sparsity and accurately tracks time-varying sparse signals. Two case studies using real-world data were used to experimentally verify the performance of the proposed method. In the first experiment, the time-domain waveform of a real-world piano recording was reconstructed using only 25% of originally sampled data. In the second experiment, incomplete satellite data of the ozone distribution in the atmosphere was accurately tracked and recovered. In both cases, the sparsity of the signals in the Fourier domain (first experiment) and the DCT domain (second experiment) were exploited to show the suitability of the proposed method for tracking dynamic sparse signals. On basis of these two experiments it was shown that the proposed method outperformed both the traditional Kalman filter and a benchmark greedy method.

Acknowledgment

This work is partly supported by the University Defence Research Centre (UDRC), UK. The authors would also like to acknowledge the European Commission for funding SmartEN ITN (Grant No. 238726) under the Marie Curie ITN FP7 programme.

REFERENCES

[1] E. Candes, 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.

[2] D. Angelosante, G. Giannakis, and E. Grossi, "Compressed sensing of time-varying signals," in *International Conference on Digital Signal Processing*, jul. 2009, pp. 1–8.

[3] R. Chartrand, "Exact reconstruction of sparse signals via nonconvex minimization," *Signal Processing Letters, IEEE*, vol. 14, no. 10, pp. 707–710, oct. 2007.

[4] M. Wakin, J. Laska, M. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. Kelly, and R. Baraniuk, "An architecture for compressive imaging," in *IEEE International Conference on Image Processing*, oct. 2006, pp. 1273–1276.

[5] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *IEEE Trans. Signal Process.*, vol. 56, no. 6, pp. 2346–2356, jun. 2008.

[6] G. Mileounis, B. Babadi, N. Kalouptsidis, and V. Tarokh, "An adaptive greedy algorithm with application to nonlinear communications," *Signal Processing, IEEE Transactions on*, vol. 58, no. 6, pp. 2998–3007, 2010.

[7] J. Jin, Y. Gu, and S. Mei, "A stochastic gradient approach on compressive sensing signal reconstruction based on adaptive filtering framework," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 4, no. 2, pp. 409–420, 2010.

[8] Y. Chen, Y. Gu, and A. Hero, "Sparse LMS for system identification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2009, pp. 3125–3128.

[9] M. E. Tipping, "The relevance vector machine," 2000.

[10] D. Wipf and B. Rao, "Sparse Bayesian learning for basis selection," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2153–2164, aug. 2004.

[11] N. Vaswani, "Kalman filtered compressed sensing," in *IEEE International Conference on Image Processing*, oct. 2008, pp. 893–896.

[12] A. Carmi, P. Gurfil, and D. Kanevsky, "Methods for sparse signal recovery using Kalman filtering with embedded pseudo-measurement norms and quasi-norms," *IEEE Trans. Signal Process.*, vol. 58, no. 4, pp. 2405–2409, apr. 2010.

[13] J. Ziniel, L. Potter, and P. Schniter, "Tracking and smoothing of time-varying sparse signals via approximate belief propagation," in *Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, 2010, pp. 808–812.

[14] A. Charles, M. Asif, J. Romberg, and C. Rozell, "Sparsity penalties in dynamical system estimation," in *Conference on Information Sciences and Systems (CISS)*, 2011, pp. 1–6.

[15] E. Karseras, K. Leung, and W. Dai, "Tracking dynamic sparse signals using hierarchical Bayesian Kalman filters." to appear in ICASSP, 2013.

[16] ——, "Tracking dynamic sparse signals with Kalman filters: A less greedy approach." to appear in SampTA, 2013.

[17] M. Tipping, A. Faul *et al.*, "Fast marginal likelihood maximisation for sparse Bayesian models," in *International workshop on artificial intelligence and statistics*, jan. 2003.

[18] J. Tropp, "Greed is good: algorithmic results for sparse approximation," *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2231–2242, oct. 2004.

[19] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Trans. Inf. Theory*, vol. 55, pp. 2230–2249, 2009.

[20] NASA, "Space-based Measurements of Ozone and Air Quality," 2012, [Online] http://ozoneaq.gsfc.nasa.gov/.