ESTIMATION AND DYNAMIC UPDATING OF TIME-VARYING SIGNALS WITH SPARSE VARIATIONS

M. Salman Asif, Adam Charles, Justin Romberg, and Christopher Rozell

School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, GA, USA

ABSTRACT

This paper presents an algorithm for an ℓ_1-regularized Kalman filter. Given observations of a discrete-time linear dynamical system with sparse errors in the state evolution, we estimate the state sequence by solving an optimization algorithm that balances fidelity to the measurements (measured by the standard ℓ_2 norm) against the sparsity of the innovations (measured using the ℓ_1 norm). We also derive an efficient algorithm for updating the estimate as the system evolves. This dynamic updating algorithm uses a homotopy scheme that tracks the solution as new measurements are slowly worked into the system and old measurements are slowly removed. The effective cost of adding new measurements is a number of low-rank updates to the solution of a linear system of equations that is roughly proportional to the joint sparsity of all the innovations in the time interval of interest.

1. INTRODUCTION

In this paper we consider the classical problem of estimating a time-varying vector described by the discrete-time linear dynamical system

$$y_i = A_i x_i + e_i \qquad (1)$$
$$x_{i+1} = F_i x_i + w_i.$$

At each time step $i = 1, 2, \ldots$, we observe M measurements $y_i \in \mathbb{R}^M$ of the unknown vectors $x_i \in \mathbb{R}^N$ through the system[1] A_i; these measurements are corrupted by a noise vector e_i. The $N \times N$ matrix F_i models the expected motion of x_i to x_{i+1}; as the motion may drift from this expected behavior, we allow for an error w_i to be incorporated at each step. We will assume throughout that the motion matrices F_i are well-conditioned enough that we may confidently calculate F_i^{-1}. No such assumption is made for the A_i, which we will generally consider as underdetermined ($M < N$).

The equations in (1) are solved in the least-squares sense by the Kalman filter [1, 2]. Given the sequence of measurements $y_1, \ldots, y_k$, and starting from an initial estimate $\widehat{x}_0$ of the vector at time $i = 0$, we can estimate $x_1, \ldots, x_k$ by solving the following optimization program

$$\underset{x_1,\ldots,x_k}{\text{minimize}} \sum_{i=1}^{k} \tau_i \|x_i - F_{i-1}x_{i-1}\|_2^2 + \sum_{i=1}^{k} \lambda_i \|A_i x_i - y_i\|_2^2. \quad (2)$$

[1]For convenience, we will assume that each A_i has the same number of rows M. Everything we present can easily be generalized to the case where we have a different number of observations at each step.

The τ_i and λ_i above are regularization parameters that we tune based on our relative confidence in the measurement and prediction errors — if the error vectors e_i and w_i are independent and identically distributed Gaussian noise vectors, then optimal choices of these parameters are based on the variances of the entries in this noise vector.

The Kalman filter owes its status as one of the pillars of statistical signal processing to the fact that the solution to (2) can be computed recursively. Say we have solved (2) after k time steps, and denote the estimate of x_k at this point as $\widehat{x}_{k|k}$. Then given a new set of measurements y_{k+1}, we can efficiently update the estimate $\widehat{x}_{k+1|k+1}$ of the new current state given only the previous estimate $\widehat{x}_{k|k}$. If A_{k+1} has M rows, then the cost of this low-rank update is essentially the same as M matrix-vector multiplies. Estimates for previous values of x_i for $i < k+1$ can be updated by working backwards ("smoothing"), with each step requiring a similar low-rank computation.

In this paper, we show how ℓ_1 regularization can be incorporated into the Kalman filtering framework. The motivation for this is that there are many applications where we expect the innovations in the vector x_i to be *sparse*. Our model replaces the second equation in (1) with

$$x_{i+1} = F_i x_i + s_i \qquad (1')$$

where s_i has only a few significant entries at unknown locations which are possibly different at each time instance. In place of the optimization program (2), we solve[2]

$$\underset{x_1,\ldots,x_k}{\text{minimize}} \sum_{i=1}^{k} \tau_i \|x_i - F_{i-1}x_{i-1}\|_1 + \frac{1}{2} \sum_{i=1}^{k} \|A_i x_i - y_i\|_2^2, \quad (3)$$

and show how the solution to this program can be efficiently updated as k increases. Our algorithm maintains the optimal solution over a sliding window of time instances, and yields the jointly optimal solution given the measurements inside the frame and the estimate from the *last frame out*, which in this case is the initial estimate $\widehat{x}_0$.

There have been several previous works with the same broad goal of incorporating discrete dynamics into sparse approximation, see in particular [3] and [4]. The formulation we give in this paper (see Section 2) varies slightly from those in these references, but our most important contribution is an efficient updating scheme for moving from one solution to the next as more observations are taken. This updating scheme, while not quite as simple as in the least-square formulation, has the same recursive spirit as the original Kalman filter, and similarly reduces to a series of low-rank updates.

The paper is organized as follows. In section 2 we discuss ℓ_1-regularized inverse problems and the formulation of our proposed dynamic recovery scheme. In section 3 we present a homotopy

[2]Note that we are weighting all of the measurements the same, leaving out the λ_i (although this is an easy extension.)

based dynamic updating algorithm for our proposed scheme. In section 4 we present some experimental results to demonstrate the performance of our scheme in estimation and tracking in the presence of sparse variations and study the performance gains from the dynamic update.

2. SPARSE VARIATION ESTIMATION

We start with a brief discussion of ℓ_1 regularization for static inverse problems. Suppose we have observed $y = Ax_0 + e$, and suppose that we have knowledge that the underlying vector x_0 is sparse. One way to take advantage of this sparsity when estimating x_0 from y is by searching for the vector with smallest ℓ_1 norm that provides a reasonable explanation for the measurements. Given y, we solve the following basis pursuit denoising (BPDN) [5] problem

$$\underset{x}{\text{minimize}}\ \tau\|x\|_1 + \frac{1}{2}\|Ax - y\|_2^2. \tag{4}$$

For measurement matrices that obey certain incoherence conditions, (4) comes with a number of performance guarantees [6–10]. In particular if x_0 is sparse enough and there is no noise, (4) recovers x exactly as $\tau \to 0$ even though A is underdetermined, and the recovery is stable in the presence of noise with an appropriate choice of τ.

The optimization program (4) does not have a closed-form solution. It is typically solved using some type of iterative descent algorithm. Recently, recursive schemes based on homotopy have been proposed to update the solution to (4) when new measurements have been observed [11–13] (i.e. entries are added to y and rows are added to A) or when y is replaced with a completely new set of measurements of a closely related signal while keeping A constant [12, 14]. These algorithms are derived by working the new measurements into (or out of) the optimization program gradually, and tracking how the solution changes. The problems are formulated such that the movement from one solution to the next can be broken down into a series of linear steps, with each link traversed using a low-rank update.

Our program (3) might be considered the dynamic version of (4). We start by reformulating the ℓ_1 version of the Kalman filter in terms of the sparse innovations s_i; we will explicitly estimate the s_i and then turn these into estimates for the x_i using the recursive equation $\widehat{x}_i = F_{i-1}x_{i-1} + \widehat{s}_i$ (recall that we are working from a fixed initial estimate $\widehat{x}_0$). The sparse innovations can be written as

$$\begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_{k-2} \\ s_{k-1} \end{bmatrix} = \begin{bmatrix} I & 0 & \cdots & 0 & 0 \\ -F_1 & I & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & -F_k & I \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{k-2} \\ x_{k-1} \end{bmatrix} - \begin{bmatrix} F_0x_0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$
$$\equiv \mathbf{s} = \mathbf{F}\mathbf{x} - \mathbf{u}. \tag{5}$$

Note that

$$\mathbf{F}^{-1} = \begin{bmatrix} I & 0 & 0 & \cdots & 0 & 0 \\ F_1 & I & 0 & \cdots & 0 & 0 \\ F_2F_1 & F_2 & I & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ F_{k\,\ldots\,1} & F_{k\,\ldots\,2} & F_{k\,\ldots\,3} & \cdots & F_k & I \end{bmatrix},$$

where $F_{k\,\ldots\,l} = F_kF_{k-1}\cdots F_l$. Since $\mathbf{x}$ can be written as $\mathbf{x} =$

$\mathbf{F}^{-1}(\mathbf{s} + \mathbf{u})$, we have the following equalities

$$\mathcal{A}\mathbf{x} = \mathcal{A}\mathbf{F}^{-1}\mathbf{s} + \mathcal{A}\mathbf{F}^{-1}\mathbf{u},$$

$$\sum_{i=1}^{k}\|A_ix_i - y_i\|_2^2 = \|\mathcal{A}\mathbf{x} - \mathbf{y}\|_2^2 = \|\mathcal{A}\mathbf{F}^{-1}\mathbf{s} - (\mathbf{y} - \mathcal{A}\mathbf{F}^{-1}\mathbf{u})\|_2^2,$$

where

$$\mathcal{A} = \begin{bmatrix} A_1 & 0 & \cdots & 0 \\ 0 & A_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_k \end{bmatrix} \text{ and } \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \end{bmatrix}.$$

Therefore, we can write the optimization problem in (3) with variables $x_1, \ldots, x_k$, in the form of following optimization problem with variables $s_1, \ldots, s_k$,

$$\underset{s_1,\ldots,s_k}{\text{minimize}} \sum_{i=1}^{k}\tau_i\|s_i\|_1 + \frac{1}{2}\|\mathcal{A}\mathbf{F}^{-1}\mathbf{s} - \widetilde{\mathbf{y}}\|_2^2, \tag{6}$$

where $\widetilde{\mathbf{y}} = \mathbf{y} - \mathcal{A}\mathbf{F}^{-1}\mathbf{u}$, and $\mathbf{u}$ is as in (5). The optimization problem in (6) is a *weighted* BPDN problem. We can solve this problem to get the sparse solution $\widehat{\mathbf{s}}$, which in turn gives us an estimate for $\mathbf{x}$.

Note that the size of the problem in (6) increases as we add new measurements in the working set $\mathcal{A}$. We will work on a sliding window, considering a fixed number P of time instances jointly. The desired length and instances of working system can be chosen by appropriately selecting the indices in (3) and (6). For example, suppose we want to keep the size of working system fixed at $PM \times PN$, and we have solved (6) at time P. Now we want to add M new measurement to the system and remove the oldest M measurements. Therefore, we solve (6) with $i = 2, \ldots, P + 1$. The main question here is whether we can quickly update the solution for (6) with index set $i = 2, \ldots, P + 1$ given a previously calculated solution for $i = 1, \ldots, P$, instead of requiring to solve it from scratch. Fortunately, the answer to this question is affirmative, and in the next section we devise a dynamic updating scheme to perform this task.

3. DYNAMIC UPDATING ALGORITHM

In this section we present an algorithm to dynamically update the solution of (3) and (6) as measurements are added or removed. To perform this task, we formulate a homotopy path between the original and updated solutions, similar to the ones used in [12], and demonstrate how this path can be traversed.

Suppose we have solved (6) for time indices $i = 1, \ldots, P$, and call the estimate for the innovations $\widehat{\mathbf{s}}$ and the corresponding state estimate $\widehat{\mathbf{x}} = \mathbf{F}^{-1}(\widehat{\mathbf{s}} + \mathbf{u})$ (which solves (3)). We break down the shift to the new time interval of interest to $i = 2, \ldots, P + 1$ into a number of small steps, each with a predictable effect on the solution. We first remove the M rows corresponding to A_1 and N columns corresponding to the first innovation s_1 from $\mathcal{A}\mathbf{F}^{-1}$. Following this, N new columns corresponding to the new innovation s_{P+1} are appended to $\mathcal{A}\mathbf{F}^{-1}$. Finally, a new set of M rows are added corresponding to A_{P+1}. Below we describe how to track the solution $\widehat{\mathbf{s}}$ to our ℓ_1 minimization program as the system undergoes these transformations. These paths between solutions will break down into a series of linear segments, each of which can be traversed very quickly. The computational cost of the algorithm scales with the number of segments in the path, which in turn is determined by the number of active elements in the set of innovations. The update will be efficient

if state solution changes only at small number of locations over the course of its evolution.

Denote the current state solution as $\widehat{\mathbf{x}} = [\widehat{x}_1, \widehat{x}_2, \cdots, \widehat{x}_P]$. Our first claim is that if we remove A_1 and x_1 from the system and solve (3) for the index set $i = 2, \ldots, P$, using $F_1 \widehat{x}_1$ as the initial estimate of x_2, then the solution remains as $[\widehat{x}_2, \cdots, \widehat{x}_P]$. This property can be verified by looking at the following optimality conditions of (3), which hold for all $i = 1, \ldots, P$,

$$A_i^T(A_i x_i - y_i) + \sum_{j=i+1}^{P-1} F_i^T \cdots F_{j-1}^T A_j^T (A_j x_j - y_j) = -\tau u_i$$

$$u_i(k) \in \begin{cases} \pm 1 & \text{if} \quad [x_i - F_{i-1} x_{i-1}](k) \neq 0 \\ [-1, 1] & \text{if} \quad [x_i - F_{i-1} x_{i-1}](k) = 0. \end{cases} \quad (7)$$

Note that the optimality conditions have a recursive relationship: if the solution $\widehat{x}_2$ is not disturbed by removing x_1 from the system, then all the subsequent solutions $\widehat{x}_3, \ldots, \widehat{x}_P$ must remain fixed. Since (7) for $i = 2$ is already satisfied with $x_1 = \widehat{x}_1$ and $x_2 = \widehat{x}_2$, our solution does not change if we remove x_1 from the system and use its estimate to form the estimate of x_2. It is clear that this argument extends beyond the first set of measurements. We can remove measurements up to any index $1 \leq j < P$ and use $F_j \widehat{x}_j$ as the initial estimate for x_{j+1} without changing the remaining solutions.

Let us now look at what happens when we add a new signal instance x_{P+1} into the system. Note that it does not affect the old solution either. This can be seen by looking at (3), where increasing the end index for prediction error term from P to $P + 1$ adds x_{P+1} into the working system. Before adding any measurements for x_{P+1}, its optimal solution will be $\widehat{x}_{P+1} = F_P \widehat{x}_P$, which implies that initial value for $\widehat{s}_{P+1} = 0$.

With the stale measurements removed and our innovation initialized to $\widehat{s}_{P+1} = 0$, we are ready for the main task of updating all of the innovation estimates $\widehat{s}_2, \ldots, \widehat{s}_{P+1}$ as we introduce the new measurements y_{P+1} into the system. We do this update using a homotopy continuation. The basic idea in the homotopy continuation is to transform a given optimization problem into a related problem, for which the solution is either already available or can be computed easily. The continuation from the transformed problem to the original problem can then be done in a series of low-complexity updates.

Let us assume that $\mathcal{A}$, $\mathbf{F}^{-1}$, $\mathbf{x}$, $\mathbf{s}$ and $\mathbf{y}$ denote the system parameters corresponding to the new time window $i = 2, \ldots, P + 1$. We compute the new $\widetilde{\mathbf{y}} = \mathbf{y} - \mathcal{A} \mathbf{F}^{-1} \mathbf{u}$, where $\mathbf{u}$ is in turn computed from $F_1 \widehat{x}_1$ in (5), and set $\widehat{\mathbf{s}}_1 = [\widehat{s}_2, \ldots, \widehat{s}_{P+1}]$. Denote first $(P-1)M$ rows of the new system matrix as $\mathbf{U}$, the last M rows of the matrix as $\mathbf{B}$, the first $(P-1)M$ elements of $\widetilde{\mathbf{y}}$ as $\widetilde{\mathbf{v}}$, and the last M elements as $\widetilde{\mathbf{w}}$. To make the exposition as simple as possible, we assume that $\tau_i = \tau$ for all i. We can now move from the old solution to the new by slowly increasing ϵ in the following homotopy formulation:

$$\min_{\mathbf{s}} \ \tau \|\mathbf{s}\|_1 + \frac{1}{2} \|\mathbf{U}\mathbf{s} - \widetilde{\mathbf{v}}\|_2^2 + \frac{1}{2} \|\mathbf{B}\mathbf{s} - \epsilon \widetilde{\mathbf{w}} - (1 - \epsilon)\mathbf{w}_{old}\|_2^2, \quad (8)$$

where $\mathbf{w}_{old} = \mathbf{B}\widehat{\mathbf{s}}_1$. At $\epsilon = 0$, $\widehat{\mathbf{s}}_1$ is the solution of (8). As we increase ϵ towards 1, the solution of (8) approaches the solution of (6) for the index set $i = 2, \ldots, P + 1$.

We traverse the solution path parameterized by ϵ in (8) using the same basic principles detailed in [12]. The solution $\mathbf{s}^\star$ of (8) at any given value of ϵ must obey the following optimality conditions

$$\mathbf{U}_\Gamma^T(\mathbf{U}\mathbf{s}^\star - \widetilde{\mathbf{v}}) + \mathbf{B}_\Gamma^T(\mathbf{B}\mathbf{s}^\star - \epsilon \widetilde{\mathbf{w}} - (1 - \epsilon)\mathbf{w}_{old}) = -\tau z \quad (9)$$

$$\|\mathbf{U}_{\Gamma^c}^T(\mathbf{U}\mathbf{s}^\star - \widetilde{\mathbf{v}}) + \mathbf{B}_{\Gamma^c}^T(\mathbf{B}\mathbf{s}^\star - \epsilon \widetilde{\mathbf{w}} - (1 - \epsilon)\mathbf{w}_{old})\|_\infty < \tau, \quad (10)$$

where Γ denotes the support of $\mathbf{s}^\star$, B_Γ is the matrix formed from the columns of B indexed by Γ, and z is the sign sequence of $\mathbf{s}^\star$ on Γ. Notice that $\mathbf{s}$ follows a piecewise linear path as ϵ varies; this path changes direction when either an element of the solution shrink to zero, or the inequality constraint (9) is violated (that is, the support of the solution changes). Suppose we are at a solution $\mathbf{s}^k$ to (8) with support Γ and signs z at some given value of $\epsilon = \epsilon_k$ between zero and one. As we increase ϵ by an infinitesimal amount to $\epsilon_k + \theta$, the solution moves to $\mathbf{s}^{k+} = \mathbf{s}^k + \theta \partial \mathbf{s}$, where

$$\partial \mathbf{s} = (\mathbf{U}^T \mathbf{U} + \mathbf{B}^T \mathbf{B})^{-1} \mathbf{B}^T (\widetilde{\mathbf{w}} - \mathbf{w}_0). \quad (11)$$

Moving in the direction of $\partial \mathbf{s}$ by increasing the step size θ, we eventually hit a *critical point* where either one of the entries in $\mathbf{s}^k$ shrinks to zero or one of the constraints in (10) becomes active (equal to τ). The smallest amount we can move ϵ so that the former is true is simply

$$\theta^- = \min_{j \in \Gamma} \left(\frac{-\mathbf{s}^k(j)}{\partial \mathbf{s}(j)} \right)_+, \quad (12)$$

where $\min(\cdot)_+$ denotes that the minimum is taken over positive arguments only. For the latter, set

$$p_k = \mathbf{U}^T(\mathbf{U}\mathbf{s}^k - \widetilde{\mathbf{y}}) + \mathbf{B}^T(\mathbf{B}\mathbf{s}^k - \epsilon_k \widetilde{\mathbf{w}} - (1 - \epsilon_k)\mathbf{w}_{old}) \quad (13a)$$

$$d_k = (\mathbf{U}^T \mathbf{U} + \mathbf{B}^T \mathbf{B}) \partial \mathbf{s} - \mathbf{B}^T(\widetilde{\mathbf{w}} - \mathbf{w}_{old}). \quad (13b)$$

We are now looking for the smallest stepsize $\Delta \epsilon$ so that $p_k(j) + \Delta \epsilon \cdot d_k(j) = \pm \tau$ for some $j \in \Gamma^c$. This is given by

$$\theta^+ = \min_{j \in \Gamma^c} \left(\frac{\tau - p_k(j)}{d_k(j)}, \frac{\tau + p_k(j)}{-d_k(j)} \right)_+. \quad (14)$$

The stepsize to the next critical point is

$$\theta = \min(\theta^+, \theta^-). \quad (15)$$

With the direction $\partial \mathbf{s}$ and stepsize θ chosen, the next critical value of ϵ and the solution at that point become

$$\epsilon_{k+1} = \epsilon_k + \theta, \quad \mathbf{s}^{k+1} = \mathbf{s}^k + \theta \partial \mathbf{s}.$$

The support for new solution $\mathbf{s}^{k+1}$ differs from Γ by one element. Let γ^- be the index for the minimizer in (12) and γ^+ be the index for the minimizer in (14). If we chose θ^- in (15), then we remove γ^- from the support Γ and the sign sequence z. If we chose θ^+ in (15), then we add γ^+ to the support, and add the corresponding sign to z. This procedure is repeated until $\epsilon = 1$.

The main computational cost at every homotopy step comes from solving a $|\Gamma| \times |\Gamma|$ system of equations to compute the direction in (11), and two matrix-vector multiplications to compute the d_k for the stepsize. Since the support changes by a single element at every homotopy step, the update direction can be computed using rank-1 update methods [15]. Furthermore, the lower block-triangular structure of $\mathbf{U}$ and $\mathbf{B}$ can be exploited to further expedite the processing and reduce the storage requirements.

4. EXPERIMENTS

In this section we present some initial experimental results to demonstrate the performance of our proposed scheme. We first analyze the performance of (3) for estimation and tracking of signals in the presence of sparse variations. Next we analyze the advantage of our proposed dynamic updating scheme over the standard BPDN homotopy solver.
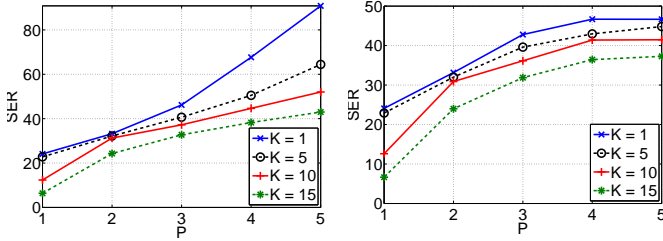
Table 1: Dynamic update comparison with BPDN homotopy.

		BPDN steps	Dynamic update steps
$P = 1$	$K = 1$	4	5
	$K = 5$	28	29
	$K = 10$	79	80
$P = 2$	$K = 1$	6	6
	$K = 5$	37	40
	$K = 10$	99	99
$P = 5$	$K = 1$	16	8
	$K = 5$	105	71
	$K = 10$	281	170
$P = 10$	$K = 1$	38	11
	$K = 5$	265	108
	$K = 10$	575	207

Fig. 1: (Left) Noiseless measurements. (Right) Noise at 30dB SNR Comparison of SER (in dB) for different values of K and P.

In our experimental setup we follow the dynamical system model in (1) and (1'), taking F_i as the identity for all i and every A_i as a random matrix whose entries are independently chosen from the standard Normal distribution. Our initial signal x_0 can be an arbitrary vector. We generate every sparse variation vector s_i independently by selecting K locations at random and assigning them amplitudes following Normal distribution. We synthesize the measurements y_i and present them as input into our algorithm along with the initial signal estimate $\widehat{x}_0$ and regularization parameter τ.

In order to evaluate the estimation and tracking performance of (3), we solved (6) for the same dynamical system using different values of P and innovation sparsity K. We initialized x_0 as a vector whose all entries are normally distributed and used zero vector as $\widehat{x}_0$. The recovery results for a system with $N = 128$ and $M = 64$ are plotted in Figure 1, with noiseless measurements (left plot) and in the presence of Gaussian noise at 30dB (right plot). The value at each point is recorded by averaging the results of 10 simulations, each run over 500 time instances of the system. The performance metric is the signal to reconstruction error ratio (SER) defined as $\text{SER} = 20 \log_{10} \frac{\|x\|_2}{\|x - \widehat{x}\|_2}$, where x denotes the original signal and $\widehat{x}$ denotes the reconstructed signal. Note that the signal estimation improves considerably as we increase P. The superior performance with larger values of P is a reflection of better signal estimation and tracking of the sparse changes in the signal dynamics.

Next we compare the cost of solving (6) from scratch using standard BPDN homotopy [16, 17] with our proposed dynamic updating algorithm. We compare the number of homotopy steps taken by each method for solving (6), using the experimental setup mentioned above, except here we initialize x_0 as a K-sparse vector. We have tabulated the average number of homotopy steps taken by both solvers for different values of P in Table 1. For small values of P both methods take approximately the same number of homotopy steps to solve (6). However, for larger values of P, the number of dynamic update steps is significantly smaller than the number of BPDN homotopy steps.

5. CONCLUSION

We have presented an ℓ_1-regularized algorithm for estimating time-varying signals with sparse prediction errors in the settings of a linear dynamical system. While a globally optimal solution would include all previous states, the computational complexity of solving such system would be overwhelming. We presented a computationally feasible algorithm which works over a sliding window of time instances, and can dynamically update the solution using homotopy updating scheme. Utilizing this algorithm, additional dynamical signal models that are not covered by the conventional Kalman filter, can be estimated in a precise and efficient manner.

6. REFERENCES

[1] R. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[2] H. W. Sorenson, "Least-squares estimation: from Gauss to Kalman," *Spectrum, IEEE*, vol. 7, no. 7, pp. 63–68, 1970.

[3] N. Vaswani, "Kalman filtered compressed sensing," *15th IEEE International Conference on Image Processing*, October 2008.

[4] D. Angelosante, S. I. Roumeliotis, and G. B. Giannakis, "Lasso-Kalman smoother for tracking sparse signals," *Forty-Third Asilomar Conference on Signals, Systems and Computers*, November 2009.

[5] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1999.

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

[7] J. Tropp, "Just relax: Convex programming methods for identifying sparse signals in noise," *IEEE Transactions on Information Theory*, vol. 52, no. 3, pp. 1030–1051, 2006.

[8] E. J. Candès and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.

[9] E. Candès, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[10] N. Meinshausen and B. Yu, "Lasso-type recovery of sparse representations for high-dimensional data," *Annals of Statistics*, vol. 37, no. 1, pp. 246–270, 2008.

[11] P. J. Garrigues and L. E. Ghaoui, "An homotopy algorithm for the Lasso with online observations," *Neural Information Processing Systems (NIPS) 21*, December 2008.

[12] M. S. Asif and J. Romberg, "Dynamic updating for ℓ_1 minimization," *IEEE Journal of Selected Topics in Signal Processing*, April 2010.

[13] ——, "Basis pursuit with sequential measurements and time varying signals," *Workshop on Computational advances in multi-sensor adaptive processing (CAMSAP)*, December 2009.

[14] ——, "Dynamic updating for sparse time varying signals," *Information Sciences and Systems, 43rd Annual Conference on (CISS)*, March 2009.

[15] Å. Björck, *Numerical Methods for Least Squares Problems*. Society for Industrial Mathematics, 1996.

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

[17] M. S. Asif and J. Romberg, "ℓ_1 Homotopy : A MATLAB toolbox for homotopy algorithms in ℓ_1 norm minimization problems," http://users.ece.gatech.edu/~sasif/homotopy.