# Sparsity Penalties in Dynamical System Estimation

Adam Charles, M. Salman Asif, Justin Romberg, Christopher Rozell
School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, Georgia 30332-0250
Email: {acharles6,sasif,jrom,crozell}@gatech.edu

***Abstract*—In this work we address the problem of state estimation in dynamical systems using recent developments in compressive sensing and sparse approximation. We formulate the traditional Kalman filter as a one-step update optimization procedure which leads us to a more unified framework, useful for incorporating sparsity constraints. We introduce three combinations of two sparsity conditions (sparsity in the state and sparsity in the innovations) and write recursive optimization programs to estimate the state for each model. This paper is meant as an overview of different methods for incorporating sparsity into the dynamic model, a presentation of algorithms that unify the support and coefficient estimation, and a demonstration that these suboptimal schemes can actually show some performance improvements (either in estimation error or convergence time) over standard optimal methods that use an impoverished model.**

***Index Terms*—Compressive Sensing, Dynamical Systems, State Estimation**

## I. Introduction

In data analysis, signal models play a crucial role in our approach to acquiring and processing signals and images. This has been especially clear in recent work where sparsity-based models have enabled dramatic improvements in the solution to linear inverse problems, especially in the context of compressive sensing (CS) for data acquisition from undersampled measurements [1], [2]. The excitement about these results is compounded by the simplicity: the algorithms often rely on solving an $\ell_1$-regularized least squares problem that can be solved with reasonable efficiency.

While many applications involve static signal estimation, many more (e.g., video) have additional statistics in the temporal signal evolution that could be exploited. For example, though each video frame may have a sparse decomposition in some basis, the regular motion in the scene is likely to induce regular changes in the coefficients from one frame to the next. In the classic dynamical systems literature, a model for a changing state vector and measurement process is often described by the following equations:

$$\begin{aligned} \boldsymbol{x}_n &= f_n(\boldsymbol{x}_{n-1}) + \boldsymbol{\nu}_n \\ \boldsymbol{y}_n &= \boldsymbol{G}_n\boldsymbol{x}_n + \boldsymbol{\epsilon}_n \end{aligned} \tag{1}$$

where $\boldsymbol{x}_n \in \mathbb{R}^N$ represents the signal of interest, $f_n(\cdot)|\mathbb{R}^N \to \mathbb{R}^N$ represents the (assumed known) evolution of the signal from time $n-1$ to $n$, $\boldsymbol{y}_n \in \mathbb{R}^M$ is a set of linear measurements of $\boldsymbol{x}_n$, $\boldsymbol{\epsilon}_n \in \mathbb{R}^M$ is the associated measurement noise, and $\boldsymbol{\nu}_n \in \mathbb{R}^N$ is our modeling error for $f_n(\cdot)$ (commonly known as the innovations). In the case where $\boldsymbol{G}_n$ is invertible ($N = M$ and the matrix has full rank), state estimation at each iteration reduces to a least squares problem. Therefore we will be particularly interested in the case of highly underdetermined measurements of the state vector (i.e., $M \ll N$), which will require us to leverage of knowledge of the state dynamics and signal structure to accurately estimate the state at each iteration.

One special, well studied, case of the system (1) assumes Gaussian statistics on both the measurement noise and the modeling error ($\boldsymbol{\epsilon}_n \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}_n)$, $\boldsymbol{\nu}_n \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_n)$), and a linear state evolution function (i.e. $f_n(\boldsymbol{x}) = \boldsymbol{F}_n\boldsymbol{x}$). The solution in this case is known as the Kalman filter [3] which estimates the optimal state at time $n$ given the current set of measurements and the previous state estimate and its covariance. While the Kalman filter is least-squares optimal in this special case, it cannot incorporate other innovations statistics or additional information about the structure of the evolving signal.

One type of signal structure which has been of particular interest recently is a low dimensional (sparse) signal structure. Compressive sensing results show that knowledge of such a structure can allow for low signal recovery error in highly undersampled systems. While such structure has been leveraged to great success in static signal processing problems, the question of how to successfully and efficiently utilize sparsity information in dynamic signal estimation remains a topic of continued research.

The framework we present in this paper seeks to capture the essence of Kalman filtering by minimizing the estimation complexity at each iteration while still accounting for both the approximately known dynamics as well as the knowledge of sparsity in the system. This approach is most similar to that taken in [4] which uses a similar explicit optimization function to account for sparse measurement noise. While not globally optimal in the sense of traditional Kalman approaches (since the proposed algorithms only use the previous state information to estimate the current state), we show that the proposed approaches can show gains in estimation error or convergence time of the estimate over both the Kalman filter or independent CS recovery at each time step (which are both operating with impoverished models due to ignoring sparsity or temporal regularity, respectively). We present in this paper three different ways to incorporate sparsity models into a dynamical system that result in estimation algorithms that are intuitive modifications to the $\ell_1$-regularized least squares problem commonly used in CS. Specifically, the three forms of sparsity in the dynamical system we propose are 1) the evolving signal $\boldsymbol{x}_n$ is sparse, 2) the error in the signal prediction $\boldsymbol{\nu}_n$ is sparse and 3) both the signal and the prediction error are sparse. For each of these three signal models we write a CS-like optimization program to solve for the signal at each time step.

## II. Background and Related Work

### A. Classical Linear State Estimation

The task of estimating the evolving state $\boldsymbol{x}_n$ has become the topic of innumerable studies and algorithms, the most notable of which is the aforementioned Kalman filter [3]. The optimal least squares estimate sought by the Kalman filter essentially requires the inversion of a large matrix with submatrices consisting of the measurement matrices $\boldsymbol{G}_n$, the dynamics matrices $\boldsymbol{F}_n$, and the identity matrix. For $n$ very small, a small number of measurements per iteration means that this matrix is underdetermined, and thus the inverse will generally be erroneous. As $n$ gets very large, this matrix becomes more and more complete (the number of columns grows as $nN$ and the number of rows grows as $n(N+M)$), which means that the error in the solution given by jointly estimating all states via a large matrix inverse fast approaches the noise floor.

The main result of the Kalman filter shows that the full matrix inverse need not be calculated if, at each iteration, we are interested only in the estimate of the current state given previous measurements. Instead each state at time $n$ can be solved for by a temporally local calculation which yields the same solution as performing the calculation intensive inverse, thereby immensely reducing the complexity in terms of the inverse problem that needs to be solved. Additionally, given the least-squares nature of the objective, the problem setup permits an analytic solution. The efficiency given both the analytic and local nature of the solution, coupled with the increasing calculation speed of matrix operations has opened the door for optimal, fast, realizable tracking in a framework common enough to be widely applicable.

The underlying assumptions the Kalman filter makes on the system (linearity and Gaussianity), however, can be quite restrictive in broader settings: Any deviation may cause the algorithm to yield erroneous solutions. Many modifications have been devised to address this shortcoming including the Extended and Unscented Kalman filters [5], [6] to address nonlinear state dynamics and many variations of robust Kalman filters have been designed to address non-Gaussian noise models [7], [8]. Since most system and noise models are difficult to account for analytically and efficiently, most of these extensions attempt to modify the resulting Kalman filter algorithm directly in order to increase the robustness of the estimation. The one main shortcoming of the Kalman filter and its derivatives is that none of the algorithms explicitly take into account any underlying signal structure (such as sparsity in the signal).

### B. Compressive Sensing

With the introduction of compressive sensing techniques, we are now equipped to efficiently deal with an entirely new set of signals: signals which are sparse in some basis. Many classical least-squares problems have benefited greatly from utilizing the inherent low dimensionality of the signal. Most notably, CS algorithms can increase the recovery capability of a sparse vector $\boldsymbol{x}$ from many fewer linear measurements $\boldsymbol{y} = \boldsymbol{G}\boldsymbol{x}$ than would otherwise be possible with least-squares solutions ($M \ll N$). The recovery, given a certain sparsity on $\boldsymbol{x}$ and compliance with certain conditions on $\boldsymbol{G}$ ($\boldsymbol{G}$ satisfies the RIP condition), is performed by regularizing the classical least squares optimization program with an $\ell_1$ norm,

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \left[ \|\boldsymbol{y} - \boldsymbol{G}\boldsymbol{x}\|_2^2 + \lambda \|\boldsymbol{x}\|_1 \right] \tag{2}$$

where $\lambda$ is a parameter which trades off data fidelity with sparsity. This optimization program is known as the Basis Pursuit De-Noising (BPDN). While many fields, such as image processing, have benefited greatly from CS results, the application to dynamic signal estimation has not been so straight-forward. In addition to the actual problem of *how* the estimation should be updated at each iteration, the question of *where* in the system sparsity could be leveraged to increase estimation performance is not well addressed. We seek in this work to explore *where* in the dynamic system it may be useful to consider sparse underlying structures as well as *how* this sparsity may be leveraged.

### C. Related Work

Previous work on this topic includes [9], [10], [11], [12], [4] and [13]. In this work we do not separate estimating the signal support from the signal values themselves such as in [9], [10]. Both work by Vaswani and Kanevsky et al. rely on modifying the existing Kalman filter Algorithm. This approach, while common in extending Kalman filtering to non-Gaussian or non-linear cases, does not explicitly attempt to find some optimal method of incorporating sparsity information. We attempt to instead start from the new model assumptions and work towards a solution using the tools brought forth by the CS results. Potter et al. and Vaswani also separate the problem of state estimation into support estimation and value estimation. While a logical avenue to explore, we do not separate these two aspects of the state. Instead we leverage the ability of $\ell_1$ regularized optimization to evaluate both the support and the values together. Additionally, in contrast to Giannakis et al. we do not smooth, i.e. we do not estimate an entire temporal range of signals together. Instead we concentrate here on the filtering aspect of state estimation, reducing the problem to a temporally local one (estimating only one state at a time). A midway solution proposed in [13] estimates the past $P$ states together, aggregating measurements for an accurate solution in the case of sparse innovations. The resulting algorithm in [13] utilizes a fast homotopy update to incorporate new measurements, speeding up the estimation process. Here we attempt to speed up the solution not by utilizing a particular solver, but by further reducing the dimension of the optimization problem to the point where we are only estimating the current state.

## III. Optimization Framework for State Estimation

The framework we present here is based on the formulation of the traditional Kalman filter as a one step optimization problem, i.e only estimates of parameters from the previous iteration can be used in the cost function. In the Kalman filter, the global solution of the state estimation problem for the system in (1) is given by the total optimization over the entire time-line

$$\{\hat{\boldsymbol{x}}_k\}_{k=0}^n = \arg\min_{\{\boldsymbol{x}_k\}_{k=0}^n} \left[ \sum_{k=0}^n \|\boldsymbol{y}_k - \boldsymbol{G}_k\boldsymbol{x}_k\|^2_{\boldsymbol{Q}_k^{-1},2} + \sum_{k=1}^n \|\boldsymbol{x}_k - \boldsymbol{F}_k\boldsymbol{x}_{k-1}\|^2_{\boldsymbol{R}_k^{-1},2} \right], \quad (3)$$

where $\|\boldsymbol{x}\|^2_{\boldsymbol{Q},2} = \boldsymbol{x}^H\boldsymbol{Q}\boldsymbol{x}$, $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$ are the covariance matrices of of the measurement noise and innovations, respectively. The Kalman filter allows us to calculate the latest state estimate $\hat{\boldsymbol{x}}_n$ from the optimization (3) locally using only the previous estimate $\hat{\boldsymbol{x}}_{n-1}$ and its covariance. The optimization program that estimates $\boldsymbol{x}_n$ alone can be written as

$$\hat{\boldsymbol{x}}_n = \arg\min_{\boldsymbol{x}_n} \left[ \|\boldsymbol{y}_n - \boldsymbol{G}_n\boldsymbol{x}_n\|^2_{\boldsymbol{Q}_n^{-1},2} + \|\boldsymbol{x}_n - \boldsymbol{F}_n\hat{\boldsymbol{x}}_{n-1}\|^2_{\boldsymbol{P}^{-1}_{n|n-1},2} \right], \quad (4)$$

where $\boldsymbol{P}_{n|n-1}$ is the estimated covariance matrix for time $n$. Both $\hat{\boldsymbol{x}}_{n-1}$ and $\boldsymbol{P}_{n|n-1}$ are parameters that are calculable iteration-to-iteration. By showing that the solution at iteration $n$ is the same for (**??**) and (**??**), the dimension of the optimization to be solved at each iteration is reduced significantly; The dimension of the solution is decreased from $nN$ to $N$. Additionally, by writing the estimation as an optimization program, we can begin to consider leveraging sparsity by applying appropriate $\ell_1$ norms in the same way that $\ell_1$ norms are introduced in static least-square cases. One encouraging application in [4] addresses a case where this formulation allows for the mitigation of sparse noise in the measurement equation. We extend this idea to directly incorporate knowledge of sparsity in the innovations and states themselves in the estimation problem.

## IV. Sparsity in the Dynamics

In previous work, the assumptions of sparsity in the system has varied. While many have assumed some measure of sparsity in the state itself [9]–[11], some have assumed knowledge of sparsity in the innovations [10] as well. Our work here takes both possibilities (sparsity in the state and innovations) and uses the framework presented in order to determine the potential gains that be realized in the context of state estimation by incorporating appropriate $\ell_1$ norms. We primarily focus on sparsity in the state evolution equation due to its relevance to specific applications, such as tracking and video. The three models we present are sparse states, sparse innovations and both sparse states and innovations. In adding the regularization terms for each case, we note that only the first order statistic of the previous estimation (the expectation) is taken into account and therefore our optimization programs are not assured to be globally optimal. This differentiates our work from (4) in that the Kalman filter which propagates second order statistics (the covariance matrix of the estimate $\boldsymbol{P}_{n|n-1}$) to obtain a globally optimal solution. As [4] points out, when deviating from the optimization problem (4), this matrix of parameters stops having an interpretation as a covariance matrix. Therefore we do not attempt to estimate second order parameters, and instead only utilize the state estimate.

### A. Sparse States

The first type of sparsity we consider is sparsity in the states only. This model still assumes that our estimate is accurate to a Gaussian random variable (e.g. $\boldsymbol{\nu}_n \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_N)$), indicating that the predicted dynamics, $f_n(\cdot)$, return a dense estimate. Such a model could potentially be considered, for instance, in a tracking problem where the number of objects to be tracked are relatively small [14]. In this case, we can add an $\ell_1$ norm over the state $\boldsymbol{x}$ to the update equation (representing our knowledge of the sparsity of the signal), resulting in

$$\begin{aligned} \hat{\boldsymbol{x}}_n = \arg\min_{\boldsymbol{x}} \big[ & \|\boldsymbol{y}_n - \boldsymbol{G}_n \boldsymbol{x}\|_2^2 + \lambda_1 \|\boldsymbol{x}\|_1 \\ & + \lambda_2 \|\boldsymbol{x} - f_n(\boldsymbol{x}_{n-1})\|_2^2 \big], \end{aligned} \tag{5}$$

where $\lambda_1$ is the sparsity parameter and $\lambda_2$ represents the ratio of the measurement variance to the innovations variance. It is important to note here that while the program (5) does not rely on linear dynamics and performs well in tracking simulations, it is has no assurance for global optimality. Thus for linear dynamics ($f_n(\boldsymbol{x}) = \boldsymbol{F}_n \boldsymbol{x}_n$) Kalman filtering still has assured optimal performance in the steady state tracking regardless of signal sparsity. This is due to the fact that the Kalman in essence is piecewise updating the solution to a larger matrix inverse problem. Given enough measurements, this matrix will be full rank, resulting in a fully determined system. Thus while our program has no assurance of obtaining a better steady-state MSE, we do expect that it will converge faster (when the Kalman filter is still underdetermined).

### B. Sparse Innovations

While including the idea of sparseness in the state is useful during convergence, there is no apparent gain in the steady state MSE over traditional Kalman filters. Where more significant gains over the Kalman filter should be realized is in the case of sparse innovations. The Gaussian assumption is key to the derivation of the Kalman filtering equations, without which the estimate covariance matrix is not exactly and analytically cacluable (making the estimate suboptimal). The sparse innovations model leads to using the $\ell_1$ norm on the error of the prediction,

$$\hat{\boldsymbol{x}}_n = \arg\min_{\boldsymbol{x}} \|\boldsymbol{y}_n - \boldsymbol{G}_n \boldsymbol{x}\|_2^2 + \lambda \|\boldsymbol{x} - f_n(\boldsymbol{x}_{n-1})\|_1, \tag{6}$$

where $\lambda$ represents the trade off between reconstruction and sparsity. A setup of this type was initially presented in [13], only with a buffer that estimated the past $P$ states at once, effectively smoothing to an extent. In keeping with the fast-update philosophy of Kalman filtering, a homotopy algorithm was used to update states given new measurements, thereby decreasing the time for the update. What is interesting in the optimization program (6) is that under a change of variables $\boldsymbol{\nu}_n = \boldsymbol{x} - f_n(\boldsymbol{x}_{n-1})$ and given a known sparsity on the innovations, the innovations is then recoverable with CS guarantees, given the typical constraints on $\boldsymbol{G}_n$. Thus with perfect knowledge of the previous state, the new state is recoverable with the same guarantees. What is not assured is the convergence of this algorithm from an erroneous initialization to a steady-state estimation error, as would be desired from a tracking algorithm. We show from simulation that it takes more measurements to have (6) converge than either of the algorithms that utilize the state sparsity directly. While obtaining a lower error vs. per-iteration measurement number, [13] shows that when estimating the past $P$ states together, the this model permits a fast update (faster than using BPDN directly) using homotopy steps.

### C. Sparse States and Sparse Innovations

The final case we consider in this paper is the case where both the state and the innovations are sparse. This combination is of the most interest to us due to its application to video where each image can be thought of as sparse in some basis and 'new' objects not predictable from older frames can be thought of as sparse innovations. In this case there are two forms of sparsity that can be leveraged. We can modify (6) to include the sparsity inducing term included in (5),

$$\begin{aligned} \hat{\boldsymbol{x}}_n = \arg\min_{\boldsymbol{x}} \big[ & \|\boldsymbol{y}_n - \boldsymbol{G}_n \boldsymbol{x}\|_2^2 + \lambda_1 \|\boldsymbol{x}\|_1 \\ & + \lambda_2 \|\boldsymbol{x} - f_n(\boldsymbol{x}_{n-1})\|_1 \big], \end{aligned} \tag{7}$$

where once again $\lambda_1$ trades off for sparsity in the state and $\lambda_2$ trades off for sparseness in the innovations.

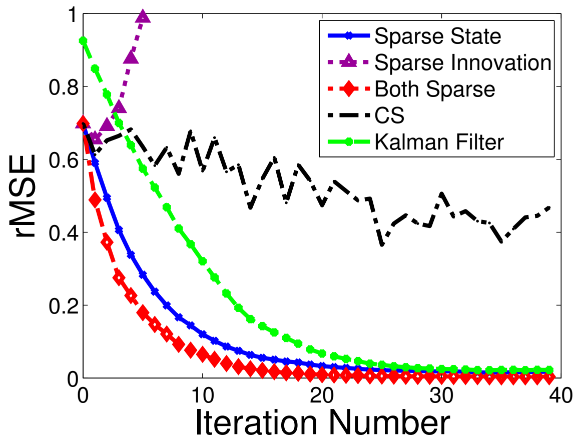


Fig. 1. By incorporating the state sparsity in the optimization program, the rMSE converges to its steady-state value faster than a traditional Kalman filter. As expected, independent BPDN performs identically at each iteration and the least matched model (sparsity in the innovations only) diverges in terms of the steady-state rMSE.

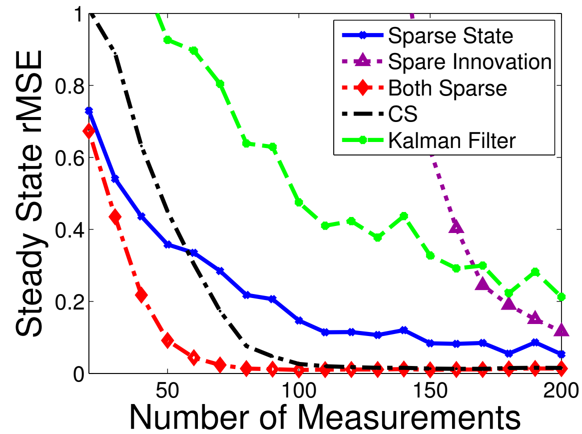


Fig. 2. Without Gaussian noise, the Kalman filter has significant trouble tracking the evolving signal, and requires more measurements than any optimization program which takes the sparsity of the signal into account, including independent BPDN. Only using sparse innovations does not outperform any model for small numbers of measurements, but converges quickly for $M > 150$.

## V. Results

We test the optimization programs on randomly generated sequences of temporally evolving signals that include sparsity in the signals and the prediction errors. First, we use a standard Gaussian innovation and compare the standard Kalman filter with the optimizations (5), (6), (7), and BPDN performed independently at each iteration (optimization (2), denoted CS in the figures) to demonstrate the utility of leveraging only the sparsity of the signal. We simulate a 20-sparse state of length 500 evolving by a permutation matrix followed by a scaling matrix (both different at each iteration, and assumed known *a-priori*) with zero-mean, 0.001 variance Gaussian innovations. A Gaussian random matrix (different at each step) is used to take 30 measurements at each iteration with i.i.d. zero mean, variance 0.01 measurement noise. For each optimization, $\lambda_1$ and $\lambda_2$ were chosen by performing a parameter sweep and choosing the best value. For Figure 1 and all subsequent simulations we initialize the state to the zero vector and obtain the expected behavior by averaging over 40 trials.

Figure 1 demonstrates that while the Kalman filter does indeed reach the noise floor after enough iteration, (5) does, as predicted, reach a lower relative MSE (rMSE) during the time frame where Kalman has not yet accumulated enough measurements. Due to the global suboptimality of (5) it does not reach lower steady-state rMSE. However, the tracking error is comparable to that of the Kalman filter which is an optimal solution in this case. What is interesting to note is that (7), the program that attempts to enforce sparsity in the state and the innovations, seems to outperform in both regimes: It obtains a lower steady-state rMSE in less iterations.

To show the performance with sparse innovations, we again estimate a simulated 20-sparse, 500-dimensional vector evolving with the same dynamics as used for Figure 1 with each of the optimization programs presented and compare to independent BPDN, and the Kalman filter. In this case, sparse innovations are introduced via a Poisson random variable with mean 2 (10% of the total number of active coefficients) choosing how many coefficients (chosen at random with a uniform probability over the support) will be switched. This effectively simulates a sparse change in the support of the signal. We allow the system to run for 50 iterations, and record the steady-state rMSE for a different number of random Gaussian measurements. Figure 2 shows that the number of measurements needed (e.g. rows of $\boldsymbol{G}_n$) for a given steady state tracking error when utilizing both knowledge of sparsity in the state and innovations is significantly less than using any other method. For this program, 60 measurements is sufficient to obtain an rMSE of approximately 3%, while with the same number of measurements independent CS has approximately 17% rMSE and both models which assume Gaussian innovations have much higher steady-state rMSE values.

Figure 3 shows results using an identical setup to Figure 2, only fixing the number of measurements at $M = 80$ and sweeping the mean number of coefficients changed (half the effective sparsity of $\boldsymbol{\nu}$). We see that the optimization in (7) again performs better in terms of the steady-state rMSE. Independent CS recovery performs as

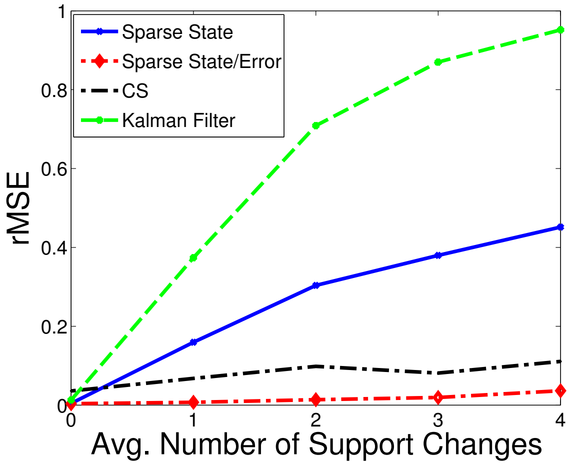


Fig. 3. The optimization taking both sparsity in the state and innovations retained the lowest steady-state rMSE for more increased innovations sparsity given a fixed number of measurements ($M = 80$). The performance for BPDN remains constant, as expected, and the performance for the models dependent on Gaussian innovations degrades quickly with additional support deviations from the expectation.

expected (the rMSE is independent of innovations), and both models using Gaussian noise obtain very high errors very quicky with the sparsity of $\boldsymbol{\nu}$. The optimization (6) is not shown here due to its inability to converge to a steady state error with only $M = 80$ measurements per iteration. It would seem that as $\boldsymbol{\nu}$ became more dense, the Gaussian model would be a better fit, but the energy over the support of $\boldsymbol{\nu}$ is on the order of the energy on the support in the state itself, so the sparsity knowledge is required to tease the two apart.

## VI. Conclusion

We have presented here a framework within which *a-priori* sparsity knowledge of a dynamically evolving state can be leveraged to estimate the current state from a small set of linear measurements at each iteration and the previous state estimate. By writing the estimate of an evolving state as a local optimization program, we are able to adapt a traditionally least-squares problem to a setting where sparsity can be directly addressed by the cost function. The three examples of sparsity in the dynamical system we introduce, and the corresponding optimization problems we use to estimate the state, show that this framework can yield improvements in both convergence rates and steady-state errors.

The formulation as it stands makes very limited assumptions on the system itself. Nowhere is the linearity of the dynamics utilized in the estimator. The algorithms do, however, require knowledge of the parameters $\lambda_1$ and $\lambda_2$ which can only be approximated based on knowledge of sparsity and noise values in the system. One avenue of future work is to determine these values more exactly based on estimated noise variance and signal/innovations sparsity. Additionally, the behavior of (6) needs to be further explored to determine why it is unable to converge in a timeframe and with measurement numbers comparable to (7) and (5). Simulations better fitting the underlying model would likely better highlight the behavior of (6).

Significant work also remains to be done to determine the extent to which these models are a substantial fit for natural signals of interest, whether more efficient algorithms can be developed to perform the estimation, how the system dynamics can be estimated if they are unknown, and whether the entire history of the dynamic process can be efficiently incorporated into the estimate in a manner similar to the Kalman filter.

## References


[1] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, Apr 2006.

[2] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, Feb 2006.

[3] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME-Journal of Basic Engineering*, vol. 82, no. D, pp. 35–45, 1960.

[4] J. Mattingley and S. Boyd, "Real-time convex optimization in signal processing," *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 50–61, May 2010.

[5] S. Haykin, "Kalman filters," in *Kalman Filtering and Neural Networks*, S. Haykin, Ed. John Wiley & Sons, Inc., 2001, pp. 1–22.

[6] E. A. Wan and R. van der Merwe, "The unscented Kalman filter," in *Kalman Filtering and Neural Networks*, S. Haykin, Ed. John Wiley & Sons, Inc., 2001, pp. 221–282.

[7] J. C. Geromel, "Optimal linear filtering under parametric uncertainty," *IEEE Transactions on Signal Processing*, vol. 47, no. 1, pp. 168–175, 1999.

[8] L. E. Ghaoui and G. Calafiore, "Robust filtering for discrete-time systems with bounded noise and parametric uncertainty," *IEEE Transactions on Automatic Control*, vol. 46, no. 7, pp. 1084–1089, Jul 2001.

[9] N. Vaswani, "Kalman filtered compressed sensing," *Proceedings of ICIP 2008*, pp. 893–896, 2008.

[10] J. Ziniel, L. C. Potter, and P. Schniter, "Tracking and smoothing of time-varying sparse signals via approximate belief propagation," *Proceeings of the Asilomar Conference on Signals, Systems and Computers*, 2010.

[11] D. Angelosante, S. I. Roumeliotis, and G. B. Giannakis, "Lasso-Kalman smoother for tracking sparse signals," *Proceeings of the Asilomar Conference on Signals, Systems and Computers*, 2009.

[12] A. Carmi, P. Gurfil, and D. Kanevsky, "Methods for sparse signal recovery using Kalman filtering pseudo-measuremennt norms and quasi-norms," *IEEE Transactions on Signal Processing*, vol. 58, no. 4, pp. 2405–2409, Apr 2010.

[13] M. S. Asif, A. Charles, J. Romberg, and C. Rozell, "Estimation and dynamic updating of time-varying signals with sparse variations," *ICASSP*, 2011.

[14] R. Baraniuk and P. Steeghs, "Compressive radar imaging," *Proceedings of the 2007 IEEE Radar Conference*, pp. 128–133, Apr 2007.