The AKRON-Kalman Filter for Tracking Time-Varying Networks

Victor Carluccio and Nidhal Bouaynaya
Dept. of Electrical and Computer Engr.
Rowan University
Glassboro, NJ, U.S.A.
vmcarluccio@gmail.com
bouaynaya@rowan.edu

Gregory Ditzler
Dept. of Electrical and Computer Engr.
University of Arizona
Tucson, AZ, U.S.A.
ditzler@email.arizona.edu

Hassan M. Fathallah-Shaykh
Department of Neurology
University of Alabama
at Birmingham
Birmingham, AL, U.S.A.
hfshaykh@uabmc.edu

***Abstract*— We propose the AKRON-Kalman filter for the problem of inferring sparse dynamic networks from a noisy undersampled set of measurements. Unlike the Lasso-Kalman filter, which uses regularization with the l_1-norm to find an approximate sparse solution, the AKRON-Kalman tracker uses the l_1 approximation to find the location of a "sufficient number" of zero entries that guarantees the existence of the optimal sparsest solution. This sufficient number of zeros can be shown to be exactly equal to the dimension of the kernel of an under-determined system. The AKRON-Kalman tracker then iteratively refines this solution of the l_1 problem by ensuring that the observed reconstruction error does not exceed the measurement noise level. The AKRON solution is sparser, by construction, than the Lasso solution while the Kalman tracking ensures that all past observations are taken into account to estimate the network in any given stage. The AKRON-Kalman tracker is applied to the inference of the time-varying wing-muscle genetic regulatory network of the *Drosophila Melanogaster* (fruit fly) during the embryonic, larval, pupal and adulthood phases. Unlike all previous approaches, the proposed AKRON-Kalman was able to recover all reportedly known interactions in the Flybase dataset.**

***Index Terms*—Time-varying genomic regulatory networks, compressive sensing, convex optimization, l_1-reconstruction.**

I. Introduction

Understanding the dynamical behavior of living cells from their complex genomic regulatory networks is a challenge posed in systems biology. Gene expression data [1] can be used to infer or reverse-engineer the underlying genomic network. However, most of the work on reverse-engineering genomic regulatory networks estimates one single static network from all available data, which is often collected during different cellular functions or developmental epochs. Summarizing expression data corresponding to different cellular stages into one network would be similar to characterizing a non-stationary signal by its Fourier spectrum. Static networks cannot reveal any regime-specific or key transient interactions that lead to biological changes [2].

The main challenge when inferring dynamic or time-varying genomic networks is the unavailability of multiple measurements or observations at each time step. Typically, there are only very few measurements available at each time step. These under-determined systems can, however, be overcome by using prior knowledge, such as sparsity. Sparsity is a desired constraint in many applications, including genomic regulatory networks, where a gene is typically related to only a few other genes within the network [2].

In our previous work [3], we addressed the problem of under-sampled sparse systems by proposing a new energy-weighted likelihood function that ensures the convergence of the likelihood function for under-determined systems with unknown covariance. The approach was coined Small-sample MUltivariate Regression with Covariance estimation (SMURC) and was applied to infer the wing-muscle genetic regulatory networks of the Drosophila melanogaster during the four phases of its development [3]. However, the estimated networks at every epoch used only the data in the corresponding epoch. In particular, the larval network ignored all the measurements in the previous embryonic phase, and so was the case for the subsequent stages.

In this paper, we use the Kalman filter to track the network across the different developmental epochs or cellular stages. In this formulation, the target being tracked is the set of edges between the genes and the measurements are given by the genes' expression data. In particular, the network at every stage uses all previous measurements, which could result in an improved estimation accuracy. The idea of Kalman tracking genomic regulatory networks has been pioneered in our previous work on Lasso-Kalman filtering [2], which uses a lasso-regularized Kalman filter to find a sparse solution. The optimal sparse solution is given by the l_0 norm defined as the number of non-zero elements in the vector, which is an NP-hard problem.

The novelty of this paper consists in proposing a different compressive sensing algorithm (than Lasso) to impose the sparsity constraint on the Kalman solution. Unlike the Lasso regularization, which uses the l_1 norm to approximate the sparse solution, we propose to use the l_1 norm to guess the location of the zeros in the solution and then solve for the corresponding sparse problem. From the Kernel RecONstruction (KRON) technique [4], we know that it is sufficient to find s correct zero locations, where s is the dimension of the Kernel of the under-determined system. Instead of enumerating all possible zero locations, we propose an Approximate KRON (AKRON) solution, which can be interpreted as a perturbation of the l_1 approximation to obtain a sparser solution.

We apply the proposed AKRON-Kalman tracker to recover the time-varying wing-muscle genetic regulatory network of the *Drosophila* and compare with state-of-the-art techniques in dynamic sparse recovery of this network, including the Lasso-Kalman filter [2], the SMURC paradigm [3], dynamic Bayesian networks [5] and random graph models [6].

II. The State-Space Model

Following the works in [2] and [3], we model the network dynamics using a state space model. The system equation is given by a random walk model, which results in a smooth evolution over time. The observation equation is given by a first-order differential equation. The state space model of the incoming edges for gene i can be shown to be [2]

$$\mathbf{a}_i(k+1) = \mathbf{a}_i(k) + w_i(k). \quad (1)$$

$$\mathbf{y}_i(k) = \mathbf{X}^t(k)\ \mathbf{a}_i(k) + v_i(k), \quad (2)$$

where $i = 1, \cdots, p$, p being the number of genes. $X(k)$ is the gene expression matrix at time k. $w_i(k)$ and $v_i(k)$ are the process and observation noise, respectively. These noise processes are assumed to be zero-mean Gaussian noise processes with the known covariances $Q(k)$ and $R(k)$, respectively, and uncorrelated to the state vector $\boldsymbol{a}_i(k)$. The full connectivity matrix, $A(k)$, can be recovered by simultaneous parallel recovery of its rows $\boldsymbol{a}_i^t(k)$ at every time instant k.

III. The AKRON-Kalman Filter

A. Constrained Kalman filtering

Our constraint space is the set of sparse vectors. We do not know a priori the degree of sparsity. Using the convex approximation l_1 of the l_0 norm, we first start by projecting the Kalman solution onto the set of "approximately sparse vectors by solving the following convex optimization problem:

$$\hat{\boldsymbol{a}} = \underset{\boldsymbol{a}}{\operatorname{argmin}}(1-\lambda)\|\boldsymbol{a}_{KF} - \boldsymbol{a}\|_2 + \lambda\|\boldsymbol{a}\|_1, \quad (3)$$

where $\boldsymbol{a}_{KF}$ is the Kalman filter estimate and λ is a parameter that controls the trade-off between sparsity and closeness to the Kalman solution. When $\lambda = 0$, (3) is simply the unconstrained Kalman estimate. When $\lambda = 1$, (3) provides the approximately sparsest solution (in terms of the l_1-norm), regardless of the Kalman estimate.

B. Perturbation of the l_1 approximation

Consider the following l_0-optimization problem, which finds the sparsest solution in a linear under-determined system

$$\text{minimize}\|\boldsymbol{x}\|_0 \text{ subject to } \boldsymbol{\Phi x} = \boldsymbol{y}, \quad (4)$$

where $\|\boldsymbol{x}\|_0$ denotes the l_0-norm of the vector $\boldsymbol{x} \in \mathbb{R}^p$, $\boldsymbol{y} \in \mathbb{R}^n$ and $n < p$. Compressive sensing theory [7] shows that, under the Restricted Isometry Property (RIP) condition on the matrix $\boldsymbol{\Phi}$, the l_1-norm solution is equivalent to the l_0-norm solution. However, it is impossible to check if the RIP condition is satisfied for a given matrix. Despite this strict condition, l_1 has been routinely used to find a sparse solution in systems of the form (4).

Consider the Kernel of $\boldsymbol{\Phi}$ defined as $Ker(\boldsymbol{\Phi}) = \{\boldsymbol{z} : \boldsymbol{\Phi z} = \boldsymbol{0}\}$. Let the dimension of $Ker(\Phi)$ be s. Without loss of generality, we can assume that $\boldsymbol{\Phi}$ is full-rank, i.e., $s = p - n$. The system in (4) admits solutions with at least s zeros [4]. We can exhaustively search all combinations of s zeros among the p entries to obtain all solutions with at least s zeros and choose the sparsest one [4]. This enumeration of s zeros among the p unknowns finds the sparsest l_0-norm solution; but at a high computational cost, as it requires $\binom{p}{s}$ enumerations. Nevertheless, finding s correct zero locations is sufficient to find the optimal sparsest solution.

We propose to find the locations of s zeros by using the l_1 approximation. The main idea is that if the l_1 solution is "close enough" to the optimal sparsest solution, then its s-smallest elements would correspond to zero locations of the optimal solution. Therefore, we compute the l_1-approximation and set the s smallest entries to zero and re-solve for the system. This approach can be viewed as a perturbation of the l_1-approximation to bring it closer to the optimal solution.

Call $\boldsymbol{x}_*$ the l_1-solution whose smallest s entries were set to zero. Resolving for the full system $\boldsymbol{\Phi x} = \boldsymbol{y}$, where $\boldsymbol{x}$ has s zeros, as located by the l_1 approximation, would reconstruct a sparser solution $\boldsymbol{x}$ that also fits the measurements noise in $\boldsymbol{y}$. Alternatively, we propose to find a sparser solution that satisfies the constraint $\|\boldsymbol{\Phi x} - \boldsymbol{y}\| \leq \epsilon$, where ϵ models the variance of the noise in the data. We do so by first checking if the initial solution $\boldsymbol{x}_*$ satisfies $\|\boldsymbol{\Phi x_*} - \boldsymbol{y}\| \leq \epsilon$. If yes, then the final solution is given by $\boldsymbol{x}_*$. If not, we iteratively set the next smallest element in $\boldsymbol{x}_*$ to zero until the constraint is satisfied. Observe that taking into account the noise in the data may lead to a sparser solution since more entries may be set to zero as long as the observation error is smaller than the threshold ϵ.

C. The AKRON-Kalman filter

The Kalman filter estimate is made sparse by incorporating the above iterative l_1 refinement technique. First, the Kalman filter estimate is found. Then, the Kalman estimate is lasso-sparsified using (3). AKRON-Kalman uses this sparse Kalman estimate as its starting off point. The $s = p - n$ smallest elements of the l_1 projection in (3) are set to zero, and then the observation error is compared to the noise level ϵ. If the error is smaller than the energy of the noise, we adopt this solution. Otherwise, the next smallest element is set to zero and the error is recalculated. The detailed AKRON-Kalman tracker algorithm is described in the below algorithm.

IV. Simulation Results

A. Synthetic data

We first evaluate the performance of the AKRON-Kalman tracker using 100 randomly generated sparse 11-gene network that evolves over 4 time points. We used 9 observations per time point. The degree of sparsity was 80%. We computed the true positive (TP), true negative (TN), false positive (FP), false negative (FN) rates, sensitivity, specificity and

accuracy. All results are shown in Table I. The sensitivity, specificity and accuracy of the AKRON-Kalman were higher than the Lasso-Kalman and classical Kalman filters.

Algorithm 1 AKRON-Kalman Tracker

1. *Initialization:* Initialize the state and estimate vectors to $a_{0|0} = \hat{a}$ and $V_{0|0} = 0$.

2. For $k = 1, \ldots, n$, do

•$Prediction$:

$$\hat{a}_{k|k-1} = a_{k-1|k-1}. \tag{5}$$

$$\mathbf{V}_{k|k-1} = \mathbf{V}_{k-1|k-1} + \mathbf{Q_k}. \tag{6}$$

•$Filtering$:

$$\mathbf{K}_k = \mathbf{V}_{k|k-1}\mathbf{X}_k(\mathbf{X}_k^t\mathbf{V}_{k|k-1}\mathbf{H}_k^t + \mathbf{R}_k)^{-1}. \tag{7}$$

$$\mathbf{a}_{k|k} = a_{k-1|k-1} + \mathbf{k}_k(y_k - \mathbf{X}_k^t a_{k-1|k-1}). \tag{8}$$

$$\mathbf{V}_{k|k} = (I - \mathbf{K}_k\mathbf{X}_k^t)\mathbf{V}_{k|k-1}. \tag{9}$$

•$Projection$: Project the estimate onto an approximate sparse space by solving the convex optimization problem in (3). Call this solution $\boldsymbol{a}_*$.

3. *Approximate Kernel RecONstruction (AKRON):* For $k = 1, \ldots, n$, do
$s = |\text{Ker}(\boldsymbol{X}^t)|$
Set s smallest entries in $\boldsymbol{a}_*$ to 0
while $\epsilon > \| \boldsymbol{X}^t\boldsymbol{a}_* - \boldsymbol{y} \|$ **do**
Set the next smallest value in $\boldsymbol{a}_*$ to zero.
Compute the error $\| \boldsymbol{X}^t\boldsymbol{a}_* - \boldsymbol{y} \|$
end while

B. Application: Time-Varying Genomic Regulatory Networks of the Drosophila Melanogaster

The application of interest is the inference of the time-varying wing-muscle genomic network of the *Drosophila Melanogaster* (fruit fly). The Drosophila microarray dataset originally consists of 4028 genes taken over 66 different time points [1]. The data includes 4 stages of the Drosophila's life: embryonic (samples 1 through 30), larval (samples 31 through 40), pupal (samples 41 through 58), and adulthood (samples 59 through 66). Flybase hosts a list of undirected gene interactions [8].

In this application, we considered a list of 11 genes that are responsible for the wing muscle development, which has been considered by many researchers before [5], [6], [9], [10]. The embryonic, pupal, and larval stages are under-sampled to 9 samples in each stage that were used in the reconstruction of the 11-gene network in each developmental epoch. All 8 time points were used in the adulthood period. To summarize, the reconstruction of the connectivity matrix uses 9 samples in the embryonic, pupal, and larval developmental stages and 8 samples in the adulthood developmental stage. The 11 gene network was reconstructed throughout each of the four developmental stages using the AKRON-Kalman algorithm. The known interactions reported in Flybase are depicted in Fig. 1.

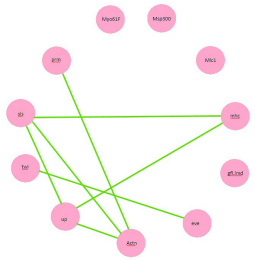


Fig. 1. Reported network from Flybase (considered as ground truth). Gene listing (Starting from the bottom going clockwise): *Actn, up, twi, sls, prm, Myo61F, Msp300, Mlc1, mhc, gfl.lmd, and eve*. Flybase does not specify the stage or the sign of the connection.

The networks reconstruction using the AKRON-Kalman tracker are shown in Figs. 2a-d. Blue edges represent a positive influence; Red edges represent negative influence, and black edges mean that the two genes influence each other in an opposite way (i.e. one positive and one negative). The AKRON-Kalman tracker was able to find every connection reported in Flybase: (*eve,twi*) appears in all four stages, (*Actn,prm*) appears in the embryonic, larval, and pupal stages, (*Actn,sls*) appears in the embryonic, larval, and pupal stages, (*Actn,up*) appears in all four stages, (*up,mhc*) appears in the embryonic, larval, and pupal stages, (*up,sls*) appears in the embryonic, larval, and pupal stages and (*sls,mhc*) appears in the embryonic and larval stages.

Table 2 lists all previous algorithms that were applied to this genetic network. Only the AKRON-Kalman, LASSO-Kalman [2], SMURC [3] and Dynamic Bayesian networks [5] considered time-varying networks; and, hence, were able to distinguish the different phases in the network. The other algorithms (minimum description length [9], random graph model [10] and nonparametric Bayesian regression [6]) assumed a stationary network, and hence it is not clear at which stage the detected connections develop. The AKRON Kalman-tracker along with the Lasso-Kalman are the only algorithms able to recover all known interactions and specify the developmental stage where this interaction occurs. Although the Lasso-Kalman also finds all reported interactions, the networks are denser (less sparse) than the AKRON-Kalman. Other algorithms are not able to detect nearly as gene interactions that are known to exist in the Drosophila as the AKRON-Kalman tracker.

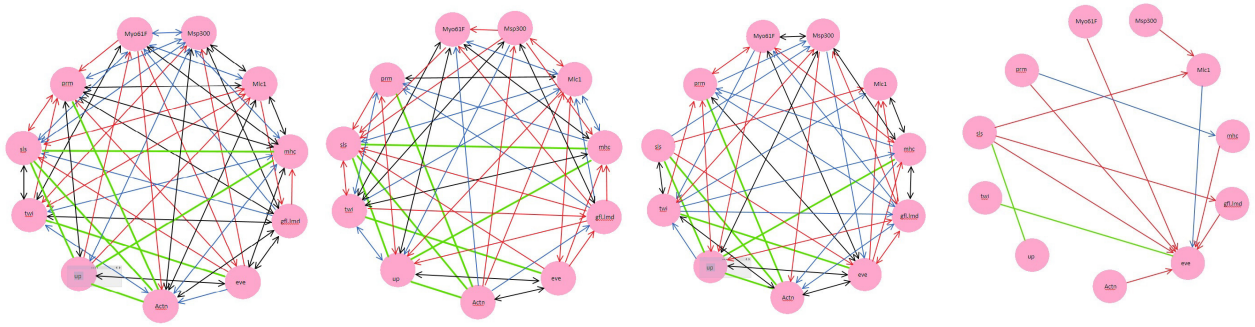
V. Discussion and Conclusion

In this paper, we formulated the inference of time-varying networks as a tracking problem, where the target consists of the network edges, which rewire, appear and disappear over time. The main difficulty in estimating time-varying networks is the lack of a sufficient number of observations per time step. However, by taking into account the sparsity of molecular networks (as a prior knowledge), we proposed the AKRON-Kalman tracker to recover the dynamic connectivity

TABLE I

PERFORMANCE ANALYSIS OF THE AKRON-KALMAN, LASSO-KALMAN AND CLASSICAL KALMAN TRACKERS

	TP	TN	FP	FN	sensitivity	specificity	accuracy
AKRON-Kalman Tracker	70.5%	17.3%	11.5%	0.7%	88.1%	98.9%	71.9%
Lasso-Kalman Tracker [2]	28.3%	12.2%	53.3%	5.5%	40.5%	83.0%	18.5%
Kalman Tracker	6.8%	16.7%	75.0%	1.3%	32.1%	82.5%	18.3%



(a) Embryonic (b) Larval (c) Pupal (d) Adult

Fig. 2. From left to right: Gene connectivity networks in the embryonic, larval, pupal and adulthood developmental stages. Red edges suppress a gene; blue edges excite a gene; black edges denote an excitation from one gene and a suppression from the other; and green edges are the connections reported in Flybase.

TABLE II

DETECTION OF THE KNOWN GENE INTERACTIONS IN FLYBASE (E: EMBRYONIC, L: LARVAL, P: PULPAL AND A: ADULTHOOD)

	(*prm,Actn*)	(*sls,mhc*)	(*mhc,up*)	(*sls,Actn*)	(*sls,up*)	(*twi,eve*)	(*up,Actn*)
AKRON-Kalman Tracker	✓ (E,L,P)	✓ (E,L)	✓ (E,L,P)	✓ (E,L,P)	✓ (E,L,P)	✓ (E,L,P,A)	✓ (E,L,P,A)
LASSO-Kalman Tracker [2]	✓ (E,L,P)	✓ (E,L)	✓ (E,L,P)	✓ (E,L,P)	✓ (E,L,P)	✓ (E,L,P,A)	✓ (E,L,P,A)
SMURC [3]	✓ (A)	✓ (A)	✓ (L)	✓ (L)	✓ (E)	✓ (P)	×
Minimum description length [9]	✓	✓	×	×	×	✓	×
Random graph model [10]	×	×	✓ (E,L,P,A)	✓ (P,A)	✓ (E,L,P,A)	×	×
Dynamic Bayesian network [5]	×	✓ (E,L,P,A)	×	×	×	×	×
Nonparametric Bayesian regression [6]	×	×	×	×	×	✓ (E)	×

of sparse networks. Using compressive sensing theory, the AKRON-Kalman tracker first computes the unconstrained Kalman solution, performs an approximate sparsification by using the l_1-norm projection, and then recursively refines this sparsification by finding the locations of the zero entries, which ensure that the observation error does not exceed the noise level in the data. The AKRON-Kalman tracker was applied to infer the wing muscle gene-regulatory network of the Drosophila Melanogaster during four developmental phases of its life cycle, and successfully identified all seven known interactions reported in Flybase.

VI. ACKNOWLEDGEMENT

This work was supported by the National Science Foundation under Grants NSF ACI-1429467 and NSF DUE-1610911.

REFERENCES

[1] M. N. Arbeitman, E. E. Furlong, F. Imam, E. Johnson, B. Null, B. S. Baker, M. A. Krasnow, M. P. Scott, R. W. Davis, and K. P. White, "Gene expression during the life cycle of Drosophila Melanogaster," *Science*, vol. 297, no. 5590, pp. 2270–2275, September 2002.

[2] J. Khan, N. Bouaynaya, and H. Fathallah-Shaykh, "Tracking of time-varying genomic regulatory networks with a Lasso-kalman smoother," *EURASIP Journal on Bioinformatics and Systems Biology*, vol. 2014, no. 1, pp. 1–27, February 2014.

[3] B. Bayar, N. Bouaynaya, and R. Sheterenberg, "SMURC: High-dimension small-sample multivariate regression with covariance estimate," *IEEE Journal of Biomedical and Health Informatics*, January 2016, under press.

[4] B. Bayar, N. Bouaynaya, and R. Shterenberg, "Kernel reconstruction: an exact greedy algorithm for compressive sensing," in *IEEE Global Conference on Signal and Information Processing*, February 2015.

[5] J. W. Robinson and A. J. Hartemink, "Learning non-stationary dynamic Bayesian networks," *The Journal of Machine Learning Research*, vol. 11, pp. 3647–3680, 2010.

[6] H. Miyashita, T. Nakamura, Y. Ida, T. Matsumoto, and T. Kaburagi, "Nonparametic Bayes-based heterogenious Drosophila Melanogaster gene regulatory network inference: T-process regression," in *Proceedings of the International Conference on Artificial Intelligence and Applications*, 2013, pp. 51–58.

[7] M. Fornasier and H. Ranhut, *Handbook of Mathematical Methods in Imaging*. Springer, 2011, vol. 1, ch. Compressive Sensing, pp. 187–228.

[8] S. Marygold, P. C. Leyland, R. L. Seal, J. L. Goodman, J. Thurmond, V. B. Strelets, and R. J. Wilson, "FLybase: improvements to the bibliography," *Nucleic acids research*, vol. 41, pp. 751–757, January 2013.

[9] W. Zhao, E. Serpedin, and E. R. Dougherty, "Inferring gene regulatory networks form time series data using minimum description length principle," *Bioinformatics*, vol. 22, no. 17, pp. 2129–2135, 2006.

[10] F. Go, S. Hanneke, W. Fu, and E. P. Xing, "Recovering temporally rewiring networks: A model based approach," in *Proceedings of the international conference of Machine Learning*, 2007, pp. 321–328.