Joint Graph Learning and Signal Recovery via Kalman Filter for Multivariate Auto-Regressive Processes

Mahmoud Ramezani-Mayiami
Dept. of ICT, University of Agder, Grimstad, Norway
Email: mahmoud.ramezani@uia.no

***Abstract*—In this paper, an adaptive Kalman filter algorithm is proposed for simultaneous graph topology learning and graph signal recovery from noisy time series. Each time series corresponds to one node of the graph and underlying graph edges express the causality among nodes. We assume that graph signals are generated via a multivariate auto-regressive processes (MAR), generated by an innovation noise and graph weight matrices. Then we relate the state transition matrix of Kalman filter to the graph weight matrices since both of them can play the role of signal propagation and transition. Our proposed Kalman filter for MAR processes, called KF-MAR, runs three main steps; prediction, update, and learn. In prediction and update steps, we fix the previously learned graph weight matrices and follow a regular Kalman algorithm for graph signal recovery. Then in the learning step, we use the last update of graph signal estimates and keep track of topology changes. Simulation results show that our proposed graph Kalman filter outperforms the available online algorithms for graph topology inference and also it can achieve the same performance of the batch method, when the number of observations increase.**

***Index Terms*—Graph signal processing, Kalman filter, Topology inference, Multivariate auto-regressive processes, Causal data network.**

I. Introduction

The explosive growth of information and data has recently raised challenging issues of data storage, statistical processing, and information inferences. The emerging field of graph signal processing (GSP) [1] initiated analysis of this huge amount of data by utilizing a graph structure, where each nodes of the graph can represent a component of the system and has a time series of data. In other words, a plethora of applications such as social networks, transportation network, wireless sensor network, and so on, can be modeled by a graph in which each node provides a time series of data which has some kind of dependencies to other ones.

One desired goal is to find the underlying dependencies or the graph topology when we are given the time series of data. A simple method to find the dependency among nodes data is to compute the correlation of two time series and if it is above a threshold, the corresponding two nodes is connected and by repeating this procedure for any pair of nodes, the underlying graph is estimated [2]. However, the correlation can not capture the directional and mediated dependencies. By a mediated dependency, we mean that if x is correlated with y and y is correlated with z, computing the correlation leads to find a correlation between x and z, while it is not true in the underlying real system. For solving this issue, partial correlation may be applied instead, while it still suffers from the problem of directionality [2]. The Granger causality [3] model may be used to find the direction of causations, where time series x is said to Granger-cause y, if knowing the previous values of x can improve the prediction accuracy of y [4]. However, the general definition of Granger causality can not capture the temporal structure of time series and we need a more specific model. The Multivariate Auto-Regressive (MAR)[1] model is one of the common tool to model the space time dependencies which is also conducive to the assessment of Granger causality. In a MAR process of order M, the current multivariate vector is a weighted superposition of M previous multivariate vectors, contaminated by an innovation noise (the mathematical model is defined in (3)).

Several researches have investigated the graph topology inference problem when the set of time series are given and the underlying process is assumed to be MAR, e.g., [5]–[7], or a similar one [8]. But in all of these researches, the set of time series is assumed to be available in advance, i.e., the proposed methods are working in batch modes. Moreover, the role of measurement noise has not been considered and they applied their methods for uncontaminated data.

However, in many applications, the data is generated over time sequentially or the computational complexity for running the algorithm in batch mode is prohibitive. Besides, data sampling comes along with a measurement noise, imposed by hardwares, e.g., sensors. To overcome the first problem, i.e., the batch mode limitations, two online graph topology inference algorithms have been proposed in [9] and [10], but the problem of noisy data is still remained. In this paper, we investigate an adaptive method to learn the underlying graph topology and recover the graph signal at the same time when we are given a set of noisy observations of MAR processes. Although Jin et. al. [11] proposed an adaptive Kalman filter for an auto-regressive model, it can not be applied for multivariate processes, like graph signals.

This paper is organized as follows; Section II reviews the basics of graph signal processing and the MAR model and at the end, we formulate the problem. The Kalman Filter

[1] in the literature of multiple time series analysis, sometimes it is also called vector auto-regressive processes (VAR).

ISBN 978-90-827970-1-5 © EURASIP 2018

for MAR processes is derived in section III. Simulations and conclusions are given in sections IV and V, respectively.

II. Problem Statement

Suppose $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a graph with the vertex set $\mathcal{V} = \{v_1, \dots, v_N\}$ and the edge set $\mathcal{E} \subset \mathcal{V}$ which captures the directed dependencies among vertices. The matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$ is a weight matrix where W_{ij} is the weight assigned to the edge connecting vertex j to i. A graph signal $\mathbf{x}[k]$ at time k is defined as below:

$$\begin{aligned} &\mathbf{x}[k] : \mathcal{V} \to \mathbb{C}^N, v_i \mapsto x_i[k] \\ &\mathbf{x}[k] = (x_1[k], x_2[k], \dots, x_N[k])^T \in \mathbb{C}^N, \end{aligned} \tag{1}$$

where $\mathbb{C}$ and T denote to the complex number and the matrix transpose, respectively. A graph Linear Shift-Invariant (LSI) filter $H(.)$ takes the input graph signal $\mathbf{x}[k]$ and gives

$$H(\mathbf{x}[k]) = \big(h_0 \mathbf{I}_N + h_1 \mathbf{W} + \cdots + h_{N_W} \mathbf{W}^{N_W}\big)\mathbf{x}[k], \tag{2}$$

for some filter coefficients $h_0, \dots\ h_{N_W}$ and a filter order N_W and $\mathbf{I}_N$ is the identity matrix of size $N \times N$.

A multivariate auto-regressive (MAR) model of order M is a LSI graph filter that generates the signal at a time k by the weighted combination of the previous M graph signals at all N nodes plus an innovation noise at time k, i.e., $\mathbf{u}[k]$ as follows

$$\mathbf{x}[k] = \sum_{m=1}^{M} \mathbf{W}_m \mathbf{x}[k-m] + \mathbf{u}[k], \tag{3}$$

where matrices $\mathbf{W}_m \in \mathbb{R}^{N \times N}$ for $m = 1, \dots, M$ are filter coefficients. We can rewrite (3) as the following element-wise expression

$$x_n[k] = \sum_{l:(n,l)\in\mathcal{E}} \sum_{m=1}^{M} w_{n,l}^{(m)} x_l[k-m] + u_n[k], n = 1, \dots, N \tag{4}$$

where $w_{n,l}^{(m)}$ is the (n,l)'th entry of the coefficient matrix $\mathbf{W}_m$. In many applications, each time series can only be affected by a subset of other time series, i.e., all nodes can not cause all other nodes. If $x_l[k]$ does not cause $x_n[k+m]$ for all $m \in \{1, \dots, M\}$, it follows $w_{n,l}^{(m)} = 0$, $\forall m$. Therefore Bolstad et. al. [5], defined the Sparse MAR Time series (SMART), considering a sparse edge set. Hereafter, we are interested in this sparse model of MAR. The problem of graph topology inference in batch mode is estimating $\mathbf{W}_m$, $m = 1, \dots, M$ when we are given the time series $\mathbf{x}[k]$ for all $k = 1, \dots, K$ in advance. If we define $\mathbf{w}_{n,l} \triangleq \big[w_{n,l}^{(1)}, w_{n,l}^{(2)}, \dots, w_{n,l}^{(M)}\big]^T$, the weight matrices can be estimated as follows

$$\begin{aligned} \{\hat{\mathbf{W}}_m\}_{m=1}^{M} = \underset{\{\mathbf{W}_m\}_{m=1}^{M}}{\operatorname{argmin}} &\sum_{k=M+1}^{K} \Big\| \mathbf{x}[k] - \sum_{m=1}^{M} \mathbf{W}_m \mathbf{x}[k-m] \Big\|_2^2 \\ &+ \lambda \sum_{n=1}^{N} \sum_{l=1}^{N} \mathbb{1}\Big(\|\mathbf{w}_{n,l}\|_1 \Big), \end{aligned} \tag{5}$$

where $\lambda > 0$ is a regularization parameter and $\mathbb{1}(.)$ is an indicator function which outputs 0 and 1 for zero and non-zero inputs, respectively. The first term of (5) ensures data fidelity and the ℓ_1-norm favors solutions which are sparse. By using the indicator function, we enforce the group sparsity. By group sparsity, we mean that if all elements of $\mathbf{w}_{n,l}$ are zero, it shows that the node l does not cause the node n for all filter taps m and it cause a more sparse solution. However, if there is even one non-zero edge $w_{n,l}^{(m)}$ between the node n and the node l, it confirms that the node n is affected by the node l. Thus, the indicator function outputs one which adds to the cost due to an additional non zero elements in filter weight matrices. Since (5) is non-convex, following the group Lasso minimization [5] with some manipulations, (5) can be rewritten as follows

$$\begin{aligned} \{\hat{\mathbf{w}}_n\}_{n=1}^{N} = \underset{\{\mathbf{w}_n\}_{n=1}^{N}}{\operatorname{argmin}} \sum_{n=1}^{N} \Big(&\sum_{k=M+1}^{K} \big(x_n[k] - \mathbf{g}^T[k]\mathbf{w}_n\big)^2 \\ &+ \lambda \sum_{l=1}^{N} \|\mathbf{w}_{n,l}\|_2 \Big), \end{aligned} \tag{6}$$

where $\mathbf{w}_n = \big[\mathbf{w}_{n,1}^T, \mathbf{w}_{n,2}^T, \dots, \mathbf{w}_{n,N}^T\big]^T$ and $\mathbf{g}$ is defined as follows

$$\mathbf{g}[k-1] \triangleq \mathrm{vec}\Big(\big[\mathbf{x}[k-1], \mathbf{x}[k-2], \dots, \mathbf{x}[k-M]\big]^T \Big), \tag{7}$$

where $\mathrm{vec}(\cdot)$ stacks columns of its input matrix in a column vector. Because of separability of the cost function across $\mathbf{w}_n$, the optimization problem (6) can be solved for each n as follows

$$\hat{\mathbf{w}}_n = \underset{\mathbf{w}_n}{\operatorname{argmin}} \sum_{k=M+1}^{K} \big(x_n[k] - \mathbf{g}^T[k]\mathbf{w}_n\big)^2 + \lambda \sum_{l=1}^{N} \|\mathbf{w}_{n,l}\|_2 . \tag{8}$$

However, in many applications, not only we can not access to all $\mathbf{x}[k],\ k = 1, \dots, K$ in advance, but also we access only to a corrupted or noisy observations $\mathbf{f}[k] = \mathbf{x}[k] + \boldsymbol{\epsilon}[k]$, where $\boldsymbol{\epsilon}[k]$ is the observation noise at time k and it is independent from graph signals and $\boldsymbol{\epsilon}[l]$ for all $l \neq k$. Therefore, in section III as the main idea of this paper, we are interested in jointly recover graph signals $\mathbf{x}[k]$ and learn the underlying graph topology in an online mode, when we are given $\mathbf{f}[k]$.

III. Kalman Filter for MAR Processes

In conventional signal processing, the Kalman Filter (KF) is mainly conceptualized by two phases: "Predict" and "Update". The predict phase manipulates the previous state estimate to produce an estimate of the state at the current time without using any new observation/measurement, so called the a priori state estimate. In the update phase, the prediction is combined with the recent observation(s) to refine the a priori estimate and yields the a posterior estimate. We can implement this algorithm whenever we have enough information about the state transition matrix, which is the mapping of the previous state to the current state. However, in this paper, the state transition matrix, which is a block matrix including $\mathbf{W}$, is not known in advance and there is a need to "learn" it from measurements. Then, our proposed algorithm, called KF for MAR processes (KF-MAR), follows three steps: predict, update, and learn. As long as new data arrives, in the prediction and update steps, graph filter matrices $\mathbf{W}_m$ are fixed and graph

ISBN 978-90-827970-1-5 © EURASIP 2018

signal states are updated, while in the learning step, we keep the updated state fixed and learn the graph filters.

The MAR model (3) can be rewritten as below

$$\mathbf{x}[k] = \mathbf{W}\mathbf{s}[k-1] + \mathbf{u}[k], \tag{9}$$

where $\mathbf{W} = [\mathbf{W}_1|\mathbf{W}_2, \dots |\mathbf{W}_M] \in \mathbb{R}^{N\times MN}$ and the state of graph signals at time $k-1$ is given as below

$$\mathbf{s}[k-1] \triangleq \big[\mathbf{x}[k-1]^T, \mathbf{x}[k-2]^T, \dots, \mathbf{x}[k-M]^T\big]^T. \tag{10}$$

If $\mathbf{s}_{n_1:n_2}$ stacks the elements of $\mathbf{s}$ from the index n_1 to the index n_2, we can rewrite (9) as follows.

$$\mathbf{s}[k]_{1:N} = \mathbf{W}\mathbf{s}[k-1] + \mathbf{u}[k], \tag{11}$$

and from (10), we have the following identity

$$\mathbf{s}[k]_{N+1:MN} = \mathbf{s}[k-1]_{1:(M-1)N}. \tag{12}$$

Therefore the state equation is expressed as follows

$$\mathbf{s}[k] = \mathbf{\Theta}\mathbf{s}[k-1] + \eta[k], \tag{13}$$

where $\boldsymbol{\eta}[k] \triangleq \big[\mathbf{u}[k-1]^T, \mathbf{u}[k-2]^T, \dots, \mathbf{u}[k-M]^T\big]^T$ and $\mathbf{\Theta} \triangleq \begin{bmatrix} [\mathbf{W}] \\ [\mathbf{I}_{(M-1)N}],\ [\mathbf{0}_{(M-1)N\times N}] \end{bmatrix}$ is the state transition matrix. Since we are given a noisy version $\mathbf{f}[k]$ of graph signals, the measurement vector is defined as $\mathbf{y}[k-1] \triangleq \big[\mathbf{f}[k-1]^T, \mathbf{f}[k-2]^T, \dots, \mathbf{f}[k-M]^T\big]^T$. Thus the measurement equation is given as below[2]

$$\mathbf{y}[k] = \mathbf{s}[k] + \boldsymbol{\xi}[k], \tag{14}$$

where $\boldsymbol{\xi}[k]$ is the measurement noise at time k, defined as follows

$$\boldsymbol{\xi}[k-1] \triangleq \big[\boldsymbol{\epsilon}[k-1]^T, \boldsymbol{\epsilon}[k-2]^T, \dots, \boldsymbol{\epsilon}[k-M]^T\big]^T. \tag{15}$$

Here, for simplicity and without loss of generality $\boldsymbol{\xi}[k]$ is assumed to be a Gaussian random process with $\boldsymbol{\xi}[k] \sim \mathcal{N}(0, \mathbf{\Lambda}_{\boldsymbol{\xi}})$ where $\mathbf{\Lambda}_{\boldsymbol{\xi}} = \sigma_\xi^2 \mathbf{I}_{MN}$. By using (13), the state is predicted as follows

$$\hat{\mathbf{s}}[k|k-1] = \hat{\mathbf{\Theta}}[k-1]\hat{\mathbf{s}}[k-1|k-1]. \tag{16}$$

where the notation $\hat{\mathbf{s}}[k|k']$ denotes the estimate of $\mathbf{s}$ at time k given all measurements $\mathbf{y}[k']$ up to and including the one at time $k' \le k$. Here, we define the error by $\mathbf{e}[k] = \mathbf{s}[k] - \hat{\mathbf{s}}[k|k]$ and the error covariance matrix as follows

$$\mathbf{P}[k|k] = E\{\mathbf{e}[k]\mathbf{e}[k]^T\}, \tag{17}$$

where $E\{\cdot\}$ is the expectation operator. From (13), we have $\mathbf{e}[k|k-1] = \hat{\mathbf{\Theta}}[k-1]\mathbf{e}[k-1] + \boldsymbol{\eta}[k]$ and by considering the zero-correlation of $\mathbf{e}[k-1]$ and $\boldsymbol{\eta}[k]$, the covariance matrix prediction is as follows

$$\mathbf{P}[k|k-1] = \hat{\mathbf{\Theta}}[k-1]\mathbf{P}[k-1|k-1]\hat{\mathbf{\Theta}}^T[k-1] + \mathbf{\Lambda}_{\boldsymbol{\eta}}, \tag{18}$$

where $\mathbf{\Lambda}_{\boldsymbol{\eta}}$ is the covariance matrix of the innovation noise and since we assumed that noises are independent, it can be expressed as $\sigma_\eta^2 \mathbf{I}_{MN}$

[2]Here, the measurement vector $\mathbf{y}[k]$ is used for the observations up to time k (including the one at k), while $\mathbf{f}[k]$ is used for the observation at time k.

It is possible to write a linear update for the a posterior estimate based on the a priori estimate and new measurement as follows

$$\hat{\mathbf{s}}[k|k] = \hat{\mathbf{s}}[k|k-1] + \mathbf{G}[k]\big(\mathbf{y}[k] - \hat{\mathbf{s}}[k|k-1]\big), \tag{19}$$

where $\mathbf{G}[k]$ is the Kalman gain, which will be derived in the following. Substituting (14) into (19) and its result into (17) gives;

$$\begin{aligned}\mathbf{P}[k|k] = \big(\mathbf{I}_{MN} - \mathbf{G}[k]\big)\mathbf{P}[k|k-1]\big(\mathbf{I}_{MN} - \mathbf{G}[k]\big)^T \\ + \sigma_\xi^2 \mathbf{G}[k]\mathbf{G}[k]^T,\end{aligned} \tag{20}$$

Here, $\sigma_{\boldsymbol{\eta}}^2$ and $\sigma_{\boldsymbol{\xi}}^2$ are constants and can be estimated via (13) and (14), respectively by $\|\hat{\mathbf{s}}[k|k-1] - \hat{\mathbf{s}}[k-1|k-1]\|_2^2$ and $\|\mathbf{y}[k] - \hat{\mathbf{s}}[k|k-1]\|_2^2$. Moreover, some methods have been proposed to estimate $\mathbf{\Lambda}_{\boldsymbol{\xi}}$ and $\mathbf{\Lambda}_{\boldsymbol{\eta}}$ in general [11] and [12].

To minimize the mean squared error, the trace of the covariance matrix is minimized as follows

$$\begin{aligned}\big(\mathbf{W}[K], \mathbf{G}[K]\big) = \underset{\mathbf{W},\mathbf{G}}{\operatorname{argmin}}\ \operatorname{tr}\big(\mathbf{P}[K|K]\big) \\ \text{s. t. } \mathbf{w}_n\text{'s are group sparse vectors,}\end{aligned} \tag{21}$$

where tr(.) is the trace operator and group sparsity was discussed in section II. This optimization problem is separable for $\mathbf{W}[K]$ and $\mathbf{G}[K]$. Taking its derivative with respect to $\mathbf{G}[K]$ and setting it to zero gives

$$\mathbf{G}[K] = \mathbf{P}[K|K-1]\big(\mathbf{P}[K|K-1] + \sigma_\xi^2 \mathbf{I}_{MN}\big)^{-1}. \tag{22}$$

Then by keeping $\mathbf{G}[K]$ fixed and following (5) and (6), (21) is minimized with respect to $\mathbf{W}$ as follows

$$\hat{\mathbf{W}}[K] = \underset{\mathbf{W}}{\operatorname{argmin}}\ J_K(\mathbf{W}), \tag{23}$$

where

$$J_K(\mathbf{W}) = \operatorname{tr}\big(\mathbf{P}[K|K]\big) + \lambda' \sum_{l=1}^{N}\sum_{n=1}^{N} \|\mathbf{w}_{n,l}\|_2, \tag{24}$$

for a regularization parameter λ'. By applying the definition of the covariance matrix (17), we have

$$\operatorname{tr}\big(\mathbf{P}[K|K]\big) = \frac{1}{K-M}\sum_{k=M+1}^{K}\left\|\mathbf{x}[k] - \sum_{m=1}^{M}\mathbf{W}_m\mathbf{x}[k-m]\right\|_2^2. \tag{25}$$

By substituting (25) into (24), we have a group Lasso batch minimization problem similar to (6) and can be rewritten separately as (8). However, we have to solve it in an online mode to adapt to the Kalman algorithm. In what follows, the gradient descent method is used to minimize the cost function $J_K(\mathbf{w}_n)$ for $n = 1, \dots, N$. Since the second term of the cost function is ℓ_1-norm of ℓ_2-norm, it is not differentiable around $\mathbf{0}$ and thus we seek one of its subdifferentials. A point $\mathbf{a} \in \mathbb{R}^N$ minimizes a non differentiable convex function $h(\mathbf{a})$ if and only if $\mathbf{0}$ is a subgradient of h at $\mathbf{a}$ [13]. Therefore, a subgradient of the ℓ_2-norm is as below

$$\nabla_{\mathbf{a}}^{\text{sub}} \|\mathbf{a}\|_2 = \begin{cases} \frac{\mathbf{a}}{\|\mathbf{a}\|_2}, & \mathbf{a} \neq \mathbf{0} \\ \mathbf{0}, & \mathbf{a} = \mathbf{0} \end{cases}, \tag{26}$$

ISBN 978-90-827970-1-5 © EURASIP 2018

and then a valid subgradient of $J_K(.)$ is given as below

$$\nabla^{\text{sub}}_{\mathbf{w}_n} J_K(\mathbf{w}_n) = \mathbf{\Phi}[K]\mathbf{w}_n - \mathbf{r}_n[K] + \lambda' g(\mathbf{w}_n), \tag{27}$$

where $g(\mathbf{w}_n) \triangleq \nabla^{\text{sub}}_{\mathbf{w}_n}\Big(\sum_{l=1}^{N} \|\mathbf{w}_{n,l}\|_2\Big)$ is the subgradient of the regularization term, $\mathbf{\Phi}[K] \triangleq \sum_{k=M+1}^{K} \mathbf{g}[k]\mathbf{g}^T[k]$ is the autocorrelation matrix of states, and $\mathbf{r}_n[K] \triangleq \sum_{k=M+1}^{K} x_n[k]\mathbf{g}[k]$ is the cross correlation. When a new observation is available, they can be updated as follows

$$\mathbf{\Phi}[K] = \mathbf{\Phi}[K-1] + \mathbf{g}[K]\mathbf{g}^T[K], \tag{28}$$

$$\mathbf{r}_n[K] = \mathbf{r}_n[K-1] + x_n[K]\mathbf{g}[K]. \tag{29}$$

Finally, $\mathbf{w}_n$ is estimated by the following standard gradient descent updates

$$\mathbf{w}_n^t = \mathbf{w}_n^{t-1} + \alpha \nabla^{\text{sub}}_{\mathbf{w}_n} J_K(\mathbf{w}_n^{t-1}), \tag{30}$$

where t denotes the gradient descent iteration index and α is the step size which guarantees the convergence of gradient descent recursions.

To conclude this section, once a new observation $\mathbf{f}[K]$ is received, we can form $\mathbf{y}[K]$ and find the a posteriori estimate of $\hat{\mathbf{s}}[K]$, so as the estimate of graph signals $\hat{\mathbf{x}}[k]$ for $k = 1, \ldots, K$. Then, we use the updated state estimate to learn the graph topology. To do this, we have to solve the optimization problem in (23) or (8) by gradient descent iterations as the inner loop, combined with the outer loop of block coordinate descent. Algorithm 1 summarizes all steps for our proposed KF-MAR. The complexity of Algorithm 1 is dominated by the matrix inversion of (22) and matrix-matrix multiplications in (18) and (20). Since the matrices are of size $MN \times MN$, the worst case computational complexity is of order $\mathcal{O}\big(M^3N^3\big)$. This computational complexity is independent of K, due to its online or recursive nature, and thus KF-MAR highly outperforms the batch mode, regarding the computational time. KF-MAR is also comparable with RRLS [9] which has overall complexity of $\mathcal{O}\big(M^2N^3\big)$ while RRLS does not recover graph signals in noisy measurements. Moreover, in many applications, we have $M \ll N \ll K$ and thus both complexities of RRLS and KF-MAR can be approximated by $\mathcal{O}\big(N^3\big)$.

Algorithm 1 Graph Kalman Filter for Multivariate Auto-Regressive Processes (KF-MAR).

Input: $\{\mathbf{f}[k]\}_{k=1}^{K}, \sigma_\xi^2, \sigma_\eta^2, \lambda, M, t_{\max}, \alpha$
Initialize: $\hat{\mathbf{w}}_n[M-1] = \mathbf{0}$, $\hat{\mathbf{s}}[M|M] = \mathbf{y}[M]$, $\mathbf{P}[M|M-1] = \sigma_\eta^2\mathbf{I}_{MN}$, $\mathbf{\Gamma}[M-1] = \delta\mathbf{I}_{MN}$ for a small δ
while $\mathbf{f}[k], k > M$ **do**
1: State vector prediction (16)
2: Covariance matrix prediction (18)
3: Compute Kalman gain (22)
4: State vector update (19)
5: Covariance matrix update (20)
6: Graph topology learning by the following loops:
for *n=1:N* **do**
for $t = 1 : t_{max}$ **do**
a) find $\nabla^{\text{sub}} J_k(\mathbf{w}_n)$ via (27)
b) update $\mathbf{w}_n^t$ via (30)
end
end
end
Output: Graph filter wights $\mathbf{W}_m$, and graph signals $\hat{\mathbf{x}}[k]$

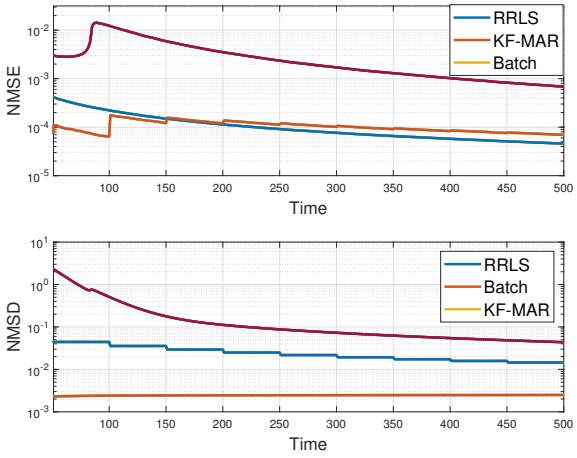

Fig. 1. NMSE and NMSD comparison for $N = 20$, $M = 4$, and $K = 500$.

IV. Simulations

We test our adaptive graph Kalman filter algorithm for synthetic data, generated by MAR processes, with the following scenario: First, an Erdős Rényi graph is generated with $N = 20$ vertices and a constant edge probability for every pair of nodes. A MAR process with order $M = 4$ is generated by drawing the non-zero coefficients of $\mathbf{W}_m, m = 1, \ldots, M$ from a Gaussian distribution, and then normalizing each matrix by its largest-magnitude eigenvalue, thus ensuring a stable MAR process. The initial graph signals $\mathbf{x}[k], k = 1, \ldots, M$ are generated from a random Gaussian $\mathcal{N}(0, 1)$ and then for $M + 1 < k \leq K$ where $K = 500$, the MAR is generated via (3), with innovation noise $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Then, the graph signals are contaminated with measurement noise $\boldsymbol{\xi}[k]$ to produce the measurements $\mathbf{y}[k]$ according to (14), where $\mathbf{\Lambda}_{\boldsymbol{\xi}} = \mathbf{I}_{MN}$.

Two performance measures are considered to compare our proposed algorithm with the conventional batch one as well as the existing online algorithm for graph-signal topology estimation. The normalized mean squared deviation $\text{NMSD}[k] = E_k\{\frac{\sum_{n=1}^{N}\|\hat{\mathbf{w}}_n[k]-\mathbf{w}_n\|_2^2}{\sum_{n=1}^{N}\|\mathbf{w}_n\|_2^2}\}$ shows the accuracy of the graph topology estimation, where $E_k\{\cdot\}$ computes the expectation up to time k. Moreover, the normalized mean squared error (NMSE) evaluates the graphs signal recovery accuracy. Because of the prohibitive computational complexity of the batch method, we run the batch method only for ten values of k, i.e., $k \in \{50, 100, \ldots, 500\}$. Fig. 1 shows the averaged results of Monte Carlo simulations over 100 trials for the three different algorithms, i.e., the batch group-Lasso [5], the Regularized Recursive Least Square (RRLS) [9], and our proposed Kalman Filter for MAR processes. Regarding

ISBN 978-90-827970-1-5 © EURASIP 2018

the NMSE, KF-MAR highly outperforms RRLS and achieves the same performance of the batch method, or even better than that, after a few observations. Moreover, our proposed algorithm has better NMSD, compared to both of the batch method and RRLS, and it converges fast for Graph topology estimation. The reason for this high performance is that the KF-MAR recovers the graph signals in each iteration while the RRLS has no signal recovery policy which forces the signal error estimation to be accumulated and affects the graph topology estimation in the next iteration.

V. Conclusion

In this paper, an algorithm for joint topology learning and graph signal recovery from noisy measurements is proposed for MAR (or VAR) processes. Our proposed method is based on the adaptive Kalman Filter algorithm. The simulation results confirm the high performance of the proposed algorithm in terms of the NMSE of graph signal recovery and the NMSD of graph topology estimation. Since the proposed algorithm is recursive, it adapts to new received data and keeps track of signal-graph changes and moreover it can apply for non-stationary processes.

ACKNOWLEDGMENT

The author would like to thank Prof. Andreas Prinz for reviewing the manuscript and his constructive editorial comments on readability of the paper.

References

[1] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, May 2013.

[2] E. D. Kolaczyk, *Statistical analysis of network data: methods and models*. Springer Science & Business Media, 2009.

[3] C. W. Granger, "Some recent development in a concept of causality," *Journal of econometrics*, vol. 39, no. 1-2, pp. 199–211, 1988.

[4] H. Lütkepohl, *New introduction to multiple time series analysis*. Springer Science & Business Media, 2005.

[5] A. Bolstad, B. D. V. Veen, and R. Nowak, "Causal network inference via group sparse regularization," *IEEE Transactions on Signal Processing*, vol. 59, no. 6, pp. 2628–2641, June 2011.

[6] Y. Shen, B. Baingana, and G. B. Giannakis, "Nonlinear structural vector autoregressive models for inferring effective brain network connectivity," *arXiv preprint arXiv:1610.06551*, 2016.

[7] J. Songsiri and L. Vandenberghe, "Topology selection in graphical models of autoregressive processes," *Journal of Machine Learning Research*, vol. 11, no. Oct, pp. 2671–2705, 2010.

[8] J. Mei and J. M. Moura, "Signal processing on graphs: Causal modeling of unstructured data," *IEEE Transactions on Signal Processing*, vol. 65, no. 8, pp. 2077–2092, 2017.

[9] B. Zaman, L. M. Lopez-Ramos, D. Romero, and B. Beferull-Lozano, "Online topology estimation for vector autoregressive processes in data networks," in *2017 IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Dec 2017, pp. 1–5.

[10] M. Ramezani-Mayiami and B. Beferull-Lozano, "Graph recursive least squares filter for topology inference in causal data processes," in *2017 IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Dec 2017, pp. 1–5.

[11] B. Jin, J. Guo, D. He, and W. Guo, "Adaptive kalman filtering based on optimal autoregressive predictive model," *GPS Solutions*, vol. 21, no. 2, pp. 307–317, 2017.

[12] A. Mohamed and K. Schwarz, "Adaptive kalman filtering for ins/gps," *Journal of geodesy*, vol. 73, no. 4, pp. 193–203, 1999.

[13] D. P. Bertsekas, *Nonlinear programming*. Athena scientific Belmont, 1999.

ISBN 978-90-827970-1-5 © EURASIP 2018