

# Dynamic Network Identification From Non-stationary Vector Autoregressive Time Series

Luis M. Lopez-Ramos, Daniel Romero, Bakht Zaman, and Baltasar Beferull-Lozano

**Abstract**— Learning the dynamics of complex systems features a large number of applications in data science. Graph-based modeling and inference underpins the most prominent family of approaches to learn complex dynamics due to their ability to capture the intrinsic sparsity of direct interactions in such systems. They also provide the user with interpretable graphs that unveil behavioral patterns and changes. To cope with the time-varying nature of interactions, this paper develops an estimation criterion and a solver to learn the parameters of a time-varying vector autoregressive model supported on a network of time series. The notion of local breakpoint is proposed to accommodate changes at individual edges. It contrasts with existing works, which assume that changes at all nodes are aligned in time. Numerical experiments validate the proposed schemes.

## D.1 Introduction

Understanding the interactions among the parts of a complex dynamic system lies at the core of data science itself and countless applications in biology, sociology, neuroscience, finance, as well as engineering realms such as cybernetics, mechatronics, and control of industrial processes. Successfully learning the presence or evolution of these interactions allows forecasting and unveils complex behaviors typically by spotting causality relations [28]. To cope with the ever increasing complexity of the analyzed systems, traditional model-based paradigms are giving way to the more contemporary data-driven perspectives, where network-based approaches enjoy great popularity due to their ability to both discern between direct and indirect causality relations as well as to provide interaction graphs amenable to intuitive human interpretation. In this context, the time-varying nature of these interactions motivates inference schemes capable of handling non-stationarity multivariate data.

Inference from multiple time series has been traditionally addressed through vector autoregressive (VAR) models [43]. To cope with non-stationarity, VAR coefficients are assumed to evolve smoothly over time [71, 72, 73], to vary according to a hidden Markov model [74], or to remain constant over time intervals separated by *structural breakpoints* [75, 76, 77, 78, 79, 42, 80]. Due to the high number of effective degrees of freedom of their models, these schemes can only satisfactorily estimate VAR coefficients if the data generating system experiences slow changes over time. To alleviate this difficulty, a natural approach is to exploit the fact that interactions among different parts of a complex system are generally *mediated*. For example, in an industrial plant where tank A is connected to B, B is connected to C, and C is not connected to A, the pressure of a fluid in a tank A affects *directly* the pressure of tank B and *indirectly* (through B) the pressure at tank C. Thus, a number of works focused on non-stationary data introduce graphs to

capture this notion of *direct* interactions, either relying on graphical models [38, 39, 14, 34] or structural equation models [26, 40]. Unfortunately, these approaches can only deal with memoryless interactions, which limits their applicability to many real-world scenarios. Schemes that do account for memory and graph structure include models based on VAR [66, 20] and structural VAR models; see [122] and references therein. However, these methods can not handle non-stationarities. To sum up, none of the aforementioned schemes identifies interaction graphs in time-varying systems with memory. To the best of our knowledge, the only exception is [44], but it can only cope with slowly changing VAR coefficients.

To alleviate these limitations, the present paper relies on a time-varying VAR (TVAR) model to propose a novel estimation criterion for non-stationary data that accounts for memory and a network structure in the interactions. The resulting estimates provide allow forecasting and *impulse response causality* analysis [43, Ch. 2]. A major novelty is the notion of *local structural breakpoint*, which captures the intuitive fact that changes in the interactions need not be synchronized across the system; in contrast to most existing works. Furthermore, a low-complexity solver is proposed to minimize the aforementioned criterion and a windowing technique is proposed to accommodate prior information on the system dynamics and reduce computational complexity.

The rest of the paper is structured as follows. Sec. D.2 introduces the model and the proposed criterion, with some practical considerations in Secs. D.2.3 and D.2.4; and Sec. D.2.5 presents an iterative solver. Numerical experiments are described in Sec. D.3 and conclusions in Sec. D.4.

## D.2 Dynamic network identification

After reviewing TVAR models and introducing the notion of time-varying causality graphs, this section proposes an estimation criterion and an iterative solver. Extensions and general considerations are provided subsequently.

### D.2.1 Time-varying interaction graphs

A multivariate time series is a collection  $\{\mathbf{y}_t\}_{t=1}^T$  of vectors  $\mathbf{y}_t := [y_{1,t}, y_{2,t}, \dots, y_{P,t}]^\top$ . The  $i$ -th (scalar) time series comprises the samples  $\{y_{i,t}\}_{t=1}^T$  and can correspond e.g. with the activity over time of the  $i$ -th region of interest in a brain network, or with the measurements of the  $i$ -th sensor in a sensor network. A customary model for multivariate time series generated by non-stationary dynamic systems is the so-called  $L$ -th order TVAR model [43, Ch. 1]:

$$\mathbf{y}_t = \sum_{\ell=1}^L \mathbf{A}_t^{(\ell)} \mathbf{y}_{t-\ell} + \boldsymbol{\varepsilon}_t \quad (\text{D.1})$$

where the matrix entries  $\{a_{ij,t}^{(\ell)}\}_{i,j \in [1,P], t \in [1,T]}$  are the model coefficients and  $\boldsymbol{\varepsilon}_{i,t}$  form the innovation process. Throughout, the notation  $[m, n]$  with  $m$  and  $n$  integers satisfying  $m \leq n$  will stand for  $\{m, m+1, \dots, n\}$ . A time-invariant VAR model is a special case of (D.1) where  $a_{ij,t}^{(\ell)} = a_{ij,t'}^{(\ell)} \forall (t, t')$ .

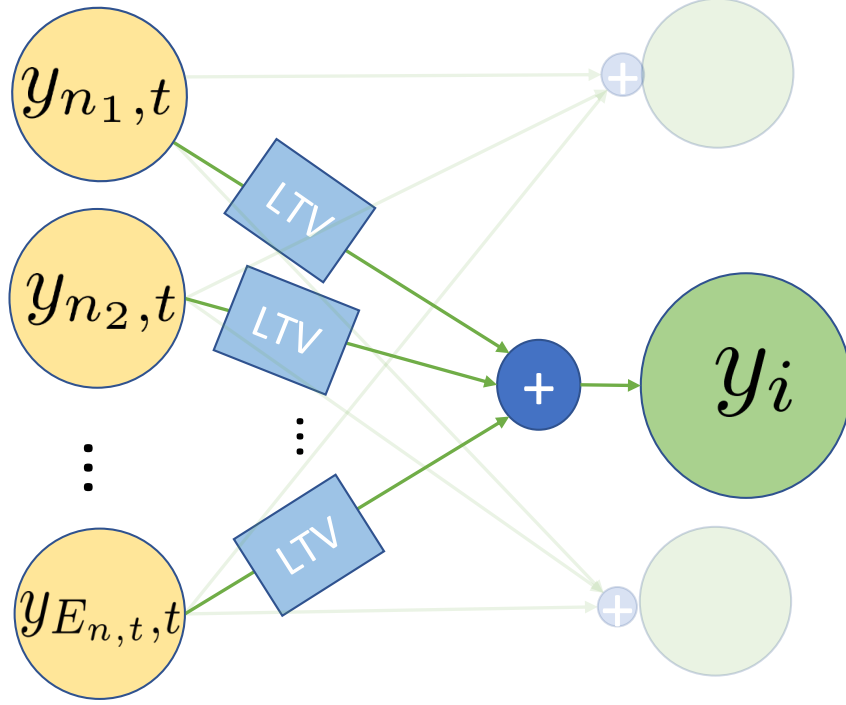


Figure D.1: Graph associated with a TVAR model.

An insightful interpretation of time-varying VAR models stems from expressing (D.1) as

$$y_{i,t} = \sum_{\ell=1}^L \sum_{j=1}^P a_{ij,t}^{(\ell)} y_{j,t-\ell} + \varepsilon_{i,t} \quad (\text{D.2a})$$

$$= \sum_{j=1}^P [y_{j,t-1}, y_{j,t-2}, \dots, y_{j,t-L}] \mathbf{a}_{ij,t} + \varepsilon_{i,t} \quad (\text{D.2b})$$

where  $\mathbf{a}_{ij,t} := [a_{ij,t}^{(1)}, a_{ij,t}^{(2)}, \dots, a_{ij,t}^{(L)}]^\top$ . From (D.2a), the  $i$ -th sequence  $\{y_{i,t}\}_{t=1}^T$  equals the innovation plus the sum of all sequences  $\{\{y_{p,t}\}_{t=1}^T\}_{p=1}^P$  after being filtered with a *linear time-varying* (LTV) filter with coefficients  $\{a_{ij,t}^{(l)}\}_{l=1}^L$ .

As described in Sec. D.1, interactions between time series are generally indirect (unmediated), which translates into many of these LTV filters being identically zero. To mathematically capture this interaction pattern, previous works consider the notion of graph associated with a time-invariant VAR process (see e.g. [66]), which is generalized next to *time-varying* VAR models (D.1). To this end, identify the  $i$ -th time series with the  $i$ -th vertex (or node) in the vertex set  $\mathcal{V} := [1, P]$  and define the time-varying edge set as  $\mathcal{E}_t := \{(i, j) \in \mathcal{V} \times \mathcal{V} : \mathbf{a}_{ij,t} \neq \mathbf{0}\}$ . Thus, each edge of this time-varying graph can be thought of as an LTV filter, as depicted in Fig. D.1.

## D.2.2 Proposed estimation criterion

The main goal of this paper is to estimate  $\{\{\mathbf{A}_t^{(\ell)}\}_{\ell=1}^L\}_{t=L+1}^T$  given  $\{\mathbf{y}_t\}_{t=1}^T$ . Without additional assumptions, reasonable estimates cannot be found because the number of unknowns is  $(T - L)P^2L$  whereas the number of samples is just  $PL$ . This difficulty is typically alleviated by assuming certain structure usually found in real-world dynamic systems. As detailed next, the structure adopted here embodies both the sparsity of causal interactions and the spatial locality of changes in those interactions.

The proposed estimation criterion is given by

$$\begin{aligned} \min_{\{\mathbf{A}_t^{(\ell)}\}} & \sum_{t=L+1}^T \left\| \mathbf{y}_t - \sum_{\ell=1}^L \mathbf{A}_t^{(\ell)} \mathbf{y}_{t-\ell} \right\|_2^2 \\ & + \sum_{(i,j)} \left( \lambda \sum_{t=L+1}^T \|\mathbf{a}_{ij,t}\|_2 + \gamma \sum_{t=L+2}^T \|\mathbf{a}_{ij,t} - \mathbf{a}_{ij,t-1}\|_2 \right) \end{aligned} \quad (\text{D.3})$$

where the first term promotes estimates that fit the data and the two regularizers in parentheses are explained next. The regularization parameters  $\lambda > 0$  and  $\gamma > 0$  can be selected through cross-validation to balance the relative weight of data and prior information (addressed in Sec. D.2.4).

The first regularizer is a group-lasso penalty that promotes edge sparsity or, equivalently, that a large number of LTV filters  $\mathbf{a}_{ij,t}$  are  $\mathbf{0}$ . As delineated in Secs. D.1 and D.2.1, this corresponds to the intuitive notion that most interactions in a complex network are indirect and therefore nodes are connected only with a small fraction of other nodes. This regularizer generalizes the one in [66] to time-varying graphs.

The second regularizer promotes estimates where the LTV filters remain constant over time except for a relatively small number of time instants  $\mathcal{T}_{i,j} := \{t : a_{ij,t}^{(\ell)} \neq a_{ij,t-1}^{(\ell)} \text{ for some } \ell\}$  denoted as *local breakpoints*. This variant of total-variation regularizer, together with the notion of local breakpoints, constitutes one of the major novelties of this paper and contrasts with the notion of structural (or global) breakpoints, defined as  $\mathcal{T} := \{t : \mathbf{A}_t^{(\ell)} \neq \mathbf{A}_{t-1}^{(\ell)} \text{ for some } \ell\}$  and adopted in the literature; see e.g. [79, 42, 75, 76, 78]. These works promote solutions with few global breakpoints, and therefore all the LTV filter estimates change simultaneously at the same time for all nodes. In contrast, this work advocates promoting solutions with few *local* breakpoints, since it is expected that changes in the underlying dynamic system take place locally. For instance, in a chemical process, closing a valve between tank A and B affects the future interactions between their pressures, but does not generally affect interactions between the pressure of tanks C and D.

## D.2.3 Data windowing

In practice, the time series are expected to evolve at a faster time scale than the underlying system that generates them. In many applications, such as control of industrial processes, the opposite would imply that the sampling rate needs to be increased. If this is the case, it may be beneficial to assume that  $\mathbf{A}_t^{(\ell)}$  remain constant within a certain window since this would decrease the number of coefficients to estimate and therefore would improve estimation performance.

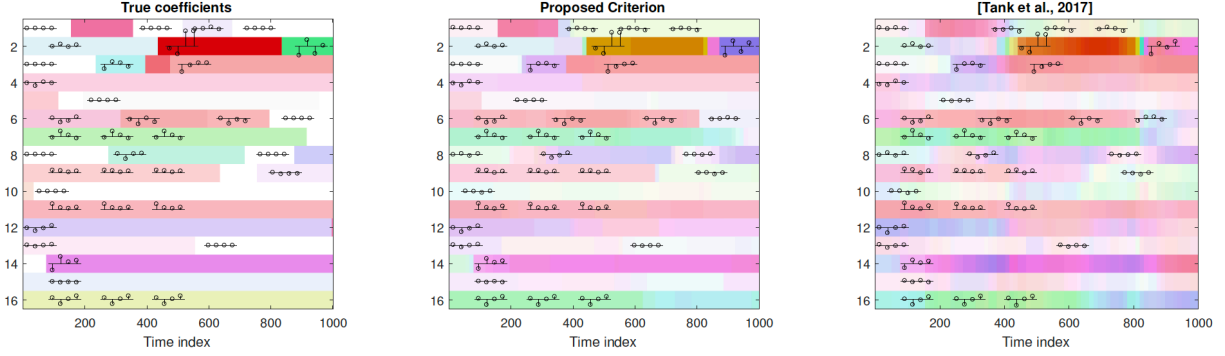


Figure D.2: Comparison between the estimates of the proposed criterion and the one in (Tank et al., 2017) [42]

To introduce this windowing technique let  $\{\mathcal{W}_n\}_{n=1}^N$  be a partition of  $[L+1, T]$  into  $N$  sub-intervals (windows), and let  $n(t)$  denote for each  $t$  the (unique) index such that  $t \in \mathcal{W}_{n(t)}$ . If  $\mathbf{A}_t^{(\ell)} = \tilde{\mathbf{A}}_{n(t)}^{(\ell)} \forall t$ , then (D.3) becomes

$$\begin{aligned} & \min_{\{\tilde{\mathbf{A}}_n^{(\ell)}\}_{n=1}^N} \sum_{t=L+1}^T \left\| \mathbf{y}_t - \sum_{\ell=1}^L \tilde{\mathbf{A}}_{n(t)}^{(\ell)} \mathbf{y}_{t-\ell} \right\|_2^2 + \sum_{(i,j)} \quad (\text{D.4}) \\ & \left( \lambda \sum_{t=L+1}^T \|\tilde{\mathbf{a}}_{ij,n(t)}\|_2 + \gamma \sum_{t=L+2}^T \|\tilde{\mathbf{a}}_{ij,n(t)} - \tilde{\mathbf{a}}_{ij,n(t-1)}\|_2 \right) \end{aligned}$$

where  $\tilde{\mathbf{a}}_{ij,t}$  is correspondingly defined in terms of  $\tilde{\mathbf{A}}_t^{(\ell)}$ . Absorbing scaling factors in the regularization parameters, (D.4) boils down to

$$\begin{aligned} & \min_{\{\tilde{\mathbf{A}}_n^{(\ell)}\}_{n=1}^N} \sum_{n=1}^N \sum_{t \in \mathcal{W}_n} \left\| \mathbf{y}_t - \sum_{\ell=1}^L \tilde{\mathbf{A}}_n^{(\ell)} \mathbf{y}_{t-\ell} \right\|_2^2 \quad (\text{D.5}) \\ & + \sum_{(i,j)} \left( \tilde{\lambda} \sum_{n=1}^N \|\tilde{\mathbf{a}}_{ij,n}\|_2 + \tilde{\gamma} \sum_{n=2}^N \|\tilde{\mathbf{a}}_{ij,n} - \tilde{\mathbf{a}}_{ij,n-1}\|_2 \right). \end{aligned}$$

Note that, while  $LP^2(T-L)$  coefficients need to be estimated in (D.3), this number reduces to  $LP^2N$  in (D.5).

Besides an improvement in the estimation performance (D.5) when the length of the windows is attuned to the dynamics of the system, it can be shown that the objective function becomes strongly convex if windows are sufficiently large, which speeds up the convergence of the algorithm in Sec. D.2.5 (convergence becomes linear). The caveat here is a loss of temporal resolution: if one wishes to detect local breakpoints and two or more changes are produced in the same LTV filter within a single window, then the algorithm will only detect at most a single breakpoint. This effect can be counteracted by applying a screening techniques along the lines of [79].

## D.2.4 Choice of parameters

Regularization parameters, in this case  $\lambda$  and  $\gamma$ , are conventionally set through cross-validation. However, such a task may be challenging when dealing with non-stationary

data. If one decides to carry out  $M$ -fold cross validation, forming  $M$  sets of consecutive samples is not appealing since the estimate of the fitting error in the validation set will become artificially high and not informative about whether the algorithm is learning changes in the VAR coefficients.

To circumvent this limitation, the proposed technique forms the aforementioned sets by skipping one out of  $M$  time samples. The estimator for the  $m$ -th fold becomes

$$\begin{aligned} & \min_{\{\tilde{\mathbf{A}}_n^{(\ell)}\}_{n=1}^N} \sum_{n=1}^N \sum_{\substack{t \in \mathcal{W}_n \\ t \bmod M \neq m}} \left\| \mathbf{y}_t - \sum_{\ell=1}^L \tilde{\mathbf{A}}_n^{(\ell)} \mathbf{y}_{t-\ell} \right\|_2^2 \\ & + \sum_{(i,j)} \left( \tilde{\lambda} \sum_{n=1}^N \|\tilde{\mathbf{a}}_{ij,n}\|_2 + \tilde{\gamma} \sum_{n=2}^N \|\tilde{\mathbf{a}}_{ij,n} - \tilde{\mathbf{a}}_{ij,n-1}\|_2 \right). \end{aligned} \quad (\text{D.6})$$

Admittedly, all vectors  $\{\mathbf{y}_t\}_t$  will still show up in all folds, but only as regressors in those folds where they are not target vectors. Indeed, this does not cause any problem from a theoretical standpoint and the performance observed in the numerical tests supports this approach.

### D.2.5 Iterative solver

This section outlines the derivation of an ADMM-based algorithm proposed to solve (D.3). Define  $\mathbf{Z} := \text{blkdiag}(\mathbf{x}_{q+1}^\top, \mathbf{x}_{q+2}^\top, \dots, \mathbf{x}_T^\top)$ , with  $\mathbf{x}_t^\top := [\mathbf{y}_{t-1}^\top \dots \mathbf{y}_{t-q}^\top]$ ;  $\mathbf{B} := [\mathbf{B}_{q+1}^\top, \dots, \mathbf{B}_T^\top]$ , with  $\mathbf{B}_t := [\mathbf{A}_t^{(1)}, \mathbf{A}_t^{(2)}, \dots, \mathbf{A}_t^{(q)}]^\top$ ; and  $\mathbf{Y} := [\mathbf{y}_{q+1}, \dots, \mathbf{y}_T]^\top$ . Then (D.3) can be rewritten as

$$\arg \min_{\mathbf{B}} \frac{1}{2} \|\mathbf{Y} - \mathbf{ZB}\|_F^2 + \lambda \Omega_{GL}(\mathbf{B}) + \gamma \Omega_{GTV}(\mathbf{B}) \quad (\text{D.7})$$

where  $\Omega_{GL}(\mathbf{B}) = \sum_{(i,j)} \sum_{t=L+1}^T \|\mathbf{a}_{ij,t}\|_2$ , and  $\Omega_{GTV}(\mathbf{B}) = \sum_{(i,j)} \sum_{t=L+1}^T \|\mathbf{a}_{ij,t+1} - \mathbf{a}_{ij,t}\|_2$ . Upon defining

$$\mathbf{D} := \begin{bmatrix} -\mathbf{I} & \mathbf{I} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} & \mathbf{I} & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \\ \mathbf{0} & & \dots & -\mathbf{I} & \mathbf{I} \end{bmatrix},$$

$\Omega_{GTV}(\mathbf{B})$  can be expressed as  $\Omega_{GL}(\mathbf{DB})$ . This allows to rewrite (D.7) along the lines of [123] for solving via ADMM

$$\begin{aligned} & \arg \min_{\mathbf{B}, \Theta, \mathbf{C}} \frac{1}{2} \|\mathbf{Y} - \mathbf{ZB}\|_F^2 + \lambda \Omega_{GL}(\Theta) + \gamma \Omega_{GL}(\mathbf{C}), \\ & \text{s.to } \mathbf{DB} = \Theta, \mathbf{B} = \mathbf{C} \end{aligned} \quad (\text{D.8})$$

The ADMM for the  $\rho$ -augmented Lagrangian with scaled dual variables  $\mathbf{U}$  and  $\mathbf{V}$

---

**Algorithm 12** ADMM solver for dynamic network ID
 

---

**Input:**  $\lambda, \gamma$ , data  $\{\mathbf{y}_t\}_{t=1}^T$ 
**Output:** matrix  $\mathbf{B}$  containing VAR coefficients

- 1: **for**  $k = 1, \dots$  until convergence **do**
  - 2:   Update  $\mathbf{B}_t$  via (D.9a)
  - 3:   **for**  $t \in [L + 1, T]$  **do**
  - 4:     **for**  $(i, j) \in [1, P]^2$  **do**
  - 5:       Update  $\mathbf{c}_{ij,t}, \theta_{ij,t-1}$  via (D.9b,D.9c)
  - 6:     **end for**
  - 7:   **end for**
  - 8:   Update  $\mathbf{U}, \mathbf{V}$  via (D.9d,D.9e)
  - 9: **end for**
- 

 computes for each iteration  $k$ :

$$\mathbf{B}^{[k+1]} := (\mathbf{Z}^\top \mathbf{Z} / \rho + \mathbf{I} + \mathbf{D}^\top \mathbf{D})^\dagger (\mathbf{Z}^\top \mathbf{Y} / \rho + \mathbf{C}^{[k]} - \mathbf{V}^{[k]} + \mathbf{D}^\top (\boldsymbol{\Theta}^{[k]} - \mathbf{U}^{[k]})) \quad (\text{D.9a})$$

$$\boldsymbol{\theta}_{ij,t}^{[k+1]} := \text{prox}_{\lambda/\rho \|\cdot\|_2} (\mathbf{b}_{ij,t}^{[k+1]} - \mathbf{b}_{ij,t-1}^{[k+1]} + \mathbf{u}_{ij,t-1}^{[k+1]}) \quad (\text{D.9b})$$

$$\mathbf{c}_{ij,t}^{[k+1]} := \text{prox}_{\lambda/\rho \|\cdot\|_2} (\mathbf{b}_{ij,t}^{[k+1]} + \mathbf{v}_{ij,t}^{[k+1]}) \quad (\text{D.9c})$$

$$\mathbf{U}^{[k+1]} := \mathbf{U}^{[k]} + (\mathbf{D}\mathbf{B}^{[k+1]} - \boldsymbol{\Theta}^{[k+1]}) \quad (\text{D.9d})$$

$$\mathbf{V}^{[k+1]} := \mathbf{V}^{[k]} + (\mathbf{B}^{[k+1]} - \mathbf{C}^{[k+1]}) \quad (\text{D.9e})$$

and it is summarized in Proc. 12. The update (D.9a) can be efficiently computed by exploiting the tri-diagonal structure of  $\mathbf{Z}$  and  $\mathbf{D}$ . The updates in (D.9b) and (D.9c) exploit the fact that the resulting prox operators are separable per  $(i, j)$  and can be expressed in terms of a group-soft-thresholding operator [124].

### D.3 Numerical experiments

A simple experiment is shown next to validate the proposed estimator. An Erdős-Rényi [13] random graph  $\mathcal{G}_0$  is generated with  $P = 4$  nodes and an edge probability of  $P_0^{(i,j)} := 0.5$  if  $i \neq j$  and  $P_0^{(i,j)} := 0$  if  $i = j$ . An  $(L = 4)$ -order TVAR model is generated, with initial VAR coefficients  $\{\mathbf{A}_{L+1}^{(\ell)}\}_{\ell=1}^L$  over  $\mathcal{G}_0$  drawn from a standard normal distribution and scaled to ensure stability [43, chapter 1]. Local breakpoints are generated at  $N_b = 100$  uniformly spaced time instants  $\mathcal{T}_b := \{t_{b1}, t_{b2}, \dots, t_{bN_b}\}$ , and for each  $t_b \in \mathcal{T}_b$  a pair of nodes  $(i_b, j_b)$  is selected uniformly at random, generating a local breakpoint at the triplet  $(t_b, i_b, j_b)$ . For each breakpoint  $b$ , the VAR coefficients  $\mathbf{a}_{i_b, j_b, t_b}$  and the edge set  $\mathcal{E}_t$  are changed as follows: if  $(i_b, j_b) \in \mathcal{E}_{t_b-1}$ ,  $\mathbf{a}_{i_b, j_b, t_b}$  is set to  $\mathbf{0}$  with probability  $P_z := 0.4$ ; otherwise, a new standard Gaussian coefficient vector  $\mathbf{a}_{i_b, j_b, t_b}$  is generated and scaled to keep stability. A realization of this TVAR process is generated by drawing  $\{\mathbf{y}_\ell\}_{\ell=1}^L$  and  $\{\boldsymbol{\varepsilon}_t\}_{t=L+1}^T$  i.i.d from a zero-mean Gaussian distribution with variance  $\sigma_\varepsilon^2 := 0.03$ .

Fig. D.2 compares the true coefficients with the estimates obtained by the proposed criterion and the one in [42]. The latter only detects global (but not local) breakpoints. The windowing described in Sec. D.2.3 selects subperiods of length  $N = 21$ , and  $\lambda$  and  $\gamma$

have been selected using the cross-validation scheme described in Sec. D.2.4, both for the proposed algorithm and the one in [42] (which only uses  $\lambda$ ).

In each subfigure, each horizontal band corresponds to a pair of nodes, and the horizontal axis represents time. The LTV impulse response vectors  $\mathbf{a}_{ij,t}/\|\mathbf{a}_{ij,t}\|$  are mapped to colors in an HSV space, being assigned similar hue if their unitary counterparts  $\mathbf{a}_{ij,t}/\|\mathbf{a}_{ij,t}\|$  are closeby. The value (brightness) is set proportional to  $\|\mathbf{a}_{ij,t}\|$ , so responses close to  $\mathbf{0}$  appear close to white, whereas impulse responses with a larger  $\ell_2$ -norm will appear in a darker color. The stems appearing between some pairs of breakpoints represent filter coefficients of  $\mathbf{a}_{ij,t}$  during the segment they lie on.

It is observed that the proposed algorithm could detect most of the local breakpoints and correctly identifies segments of stationarity. On the other hand, the competing algorithm yields a high number of false positives as expected.

## D.4 Conclusions

Dynamic networks can be identified using the notion of local breakpoints, when VAR coefficient changes appear in a small number of edges. The proposed technique involves three novelties: a regularized criterion, a windowing technique, and a cross-validation scheme. Simulation experiments encourage further research along these lines.