The Autoregressive Linear Mixture Model: a Time-Series Model for an Instantaneous Mixture of Network Processes

Addison W. Bohannon, *Member, IEEE,* Vernon J. Lawhern, *Member, IEEE,*
Nicholas R. Waytowich, *Member, IEEE,* and Radu V. Balan, *Senior Member, IEEE*

***Abstract*—Vector autoregressive models provide a simple generative model for multivariate, time-series data. The autoregressive coefficients of the vector autoregressive model describe a network process. However, in real-world applications such as macroeconomics or neuroimaging, time-series data arise not from isolated network processes but instead from the simultaneous occurrence of multiple network processes. Standard vector autoregressive models cannot provide insights about the underlying structure of such time-series data. In this work, we present the autoregressive linear mixture (ALM) model. The ALM proposes a decomposition of time-series data into co-occurring network processes that we call autoregressive components. We also present a non-convex likelihood-based estimator for fitting the ALM model and show that it can be solved using the proximal alternating linearized minimization (PALM) algorithm. We validate the ALM on both synthetic and real-world electroencephalography data, showing that we can disambiguate task-relevant autoregressive components that correspond with distinct network processes.**

***Index Terms*—time series analysis, autoregressive processes, mixture models, unsupervised learning, electroencephalography**

I. Introduction

VECTOR autoregressive models describe multivariate time-series with linear interactions among the variates at multiple time-scales. Let $(\mathbf{x}[t])_{t\in\mathbb{Z}}$ be a d-dimensional vector-valued time-series generated by a p-order autoregressive process [1],

$$\mathbf{x}[t] = \sum_{s=1}^{p} \mathbf{A}[s]\mathbf{x}[t-s] + \mathbf{n}[t]. \tag{1}$$

Here, $(\mathbf{A}[s])_{s=1,\dots,p}$ with $\mathbf{A}[s] \in \mathbb{R}^{d\times d}$ for all $s = 1,\dots,p$ are the autoregressive coefficients, and $(\mathbf{n}[t])_{t\in\mathbb{Z}}$ is the d-dimensional noise process, where $\mathbf{n}[t] \sim_{\text{iid}} \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ for some $\mathbf{\Sigma} \succeq \mathbf{0} \in \mathbb{R}^{d\times d}$. The autoregressive coefficients specify the interaction among the variates. For example, the i,j-entry of $\mathbf{A}[s]$ captures the effect of the j entry of $\mathbf{x}[t-s]$ on the i entry of $\mathbf{x}[t]$. These values could be correlated, anti-correlated, or unrelated depending on whether $A_{i,j}[s]$ is positive, negative, or zero. In this way, (1) provides a simple but powerful generative model for natural data. Accordingly, practitioners from network neuroscience and econometrics commonly use vector autoregressive models and their variants to analyze data and forecast future outcomes.

The autoregressive coefficients of a vector autoregressive model describe a single network process, a structural relationship among the variates in time. However, in many real-world applications, where time-series data are derived from observation of complex natural phenomena such as macroeconomic data [2] and neuroimaging [3]–[5], we might ask: do we believe that we observe an isolated network process? If not, how do we account for the simultaneous occurrence of other network processes? Using the example above, perhaps $x_i[t-s]$ is positively correlated with $x_j[t]$ for a particular network process and negatively correlated for another network process. If these network processes co-occur, then we will observe no correlation between the variates. If we are only interested in prediction, this subtlety may not matter, but if we want to gain insight into the process, this should be concerning.

In this work, we propose a model for the simultaneous occurrence of network processes: the autoregressive linear mixture (ALM) model. The ALM model posits a decomposition of the autoregressive coefficients into r canonical autoregressive components, *i.e.* $\mathbf{A}[s] = \sum_{j=1}^{r} c_j \mathbf{D}_j[s]$ for $s = 1,\dots,p$ with $c_j \in \mathbb{R}$ and $\mathbf{D}_j[s] \in \mathbb{R}^{d\times d}$ for all $j = 1,\dots,r$ and $s = 1,\dots,p$, so that the ALM model is defined by the recurrence relation

$$\mathbf{x}[t] = \sum_{s=1}^{p}\sum_{j=1}^{r} c_j \mathbf{D}_j[s]\mathbf{x}[t-s] + \mathbf{n}[t]. \tag{2}$$

We will denote the ALM model of order p and number of components r as ALM (p, r). The autoregressive components $\left\{(\mathbf{D}_j[s])_{s=1,\dots,p} : j = 1,\dots,r\right\}$ represent distinct network processes, and the mixing coefficients $\mathbf{c} = (c_j)_{j=1,\dots,r}$ indicate how these network processes co-occur to bring about the observed activity.

In the remainder of the manuscript, we review related time-series and mixture models. We present a non-convex likelihood-based estimator for the ALM model and propose to derive estimates using the proximal alternating linearized minimization (PALM) algorithm [6]. We characterize the performance of the PALM algorithm on simulated data. Using the PALM algorithm, we fit the ALM model to electroencephalog-

AWB, VJL, and NRW are with the Army Research Laboratory, Aberdeen Proving Ground, MD 21005, e-mail: addison.w.bohannon.civ@mail.mil, vernon.j.lawhern.civ@mail.mil, nicholas.r.waytowich.civ@mail.mil

RVB is with the Department of Mathematics and Center for Scientific Computation and Mathematical Modeling, University of Maryland, College Park, MD 20742, e-mail: rvbalan@math.umd.edu. The work of RVB was supported in part by NSF under Grant DMS-1816608.

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org provided by the author. The material includes additional details on the algorithms used and results not included in the manuscript. This material is 6MB in size.

raphy (EEG) recordings from the ISRUC-Sleep dataset (available from sleeptight.isr.uc.pt/ISRUC_Sleep/) [7]. We show that the recovered autoregressive components correspond to known network processes used for sleep-stage classification. We provide a software package for fitting the ALM model available at github.com/addisonbohannon/alm with implementations of all experiments in Sections IV and V.

II. Background

A. Related Work

In time-series analysis, and specifically econometrics, researchers have previously developed mixture models for autoregressive processes. These researchers have been interested in generative models for time-series data which feature transient behavior such as bursts and cycles. The first mixture autoregressive model is credited to Le, Martin, and Rafferty for the development of the Gaussian mixture transition distribution (GMTD) [8]. Wong and Li generalized the GMTD in developing the mixture autoregressive (MAR) model [9]. The MAR models each observation within a realization as a Gaussian mixture of autoregressive processes. Various multivariate extensions to the MAR have been proposed in Fong, *et al.* [10]; Bec, Rahbek, and Shephard [11]; Duekker, *et al.* [12]; and Kalliovirta, Meitz, and Saikkonen [13]. Fong, *et al.* propose the mixture vector autoregression (MVAR) model, which most immediately generalizes the MAR. It is this model that we use for comparison. Let $(\mathbf{x}[t])_{t\in\mathbb{Z}}$ be a random d-dimensional vector-valued time-series and $\mathcal{F}_t$ be the σ-algebra generated by observations up to time t, $\{\mathbf{x}[s] : s \leq t\}$. Also, let $(\mathbf{z}[t])_{t\in\mathbb{Z}}$ be a random time-series of iid r-dimensional categorical random variables with parameters $(c_j)_{j=1,\dots,r}$. Then, the MVAR with r components is given by

$$\begin{aligned}&\mathrm{P}\left(x_1[t] \leq x_1, \dots, x_d[t] \leq x_d \mid \mathcal{F}_{t-1}\right)\\ &\quad= \sum_{j=1}^{r} c_j \mathbf{\Phi}\left(\mathbf{\Sigma}_j^{-1/2}\left(\mathbf{x} - \sum_{s=1}^{p_j}\mathbf{D}_j[s]\mathbf{x}[t-s]\right)\right)\end{aligned} \tag{3}$$

for any $\mathbf{x} = (x_i)_{i=1,\dots,d} \in \mathbb{R}^d$, where $\mathbf{\Phi}(\mathbf{x}) = \prod_{i=1}^{d}\Phi(x_i)$ and Φ is the cumulative distribution function of the univariate normal distribution, $\mathbf{\Sigma}_j \succeq \mathbf{0}$ is the covariance of the j component, p_j is the model order of the j component, and $(\mathbf{D}_j[s])_{s=1,\dots,p_j}$ are the autoregressive coefficients of the j component.

To estimate the parameters of the MAR model, Wong and Li propose an expectation-maximization (EM) algorithm [14]. Fong, *et al.* follow suit for the MVAR model [10]. In the expectation step of the algorithm, the conditional expectation of $(\mathbf{z}[t])_{t\in\mathbb{Z}}$ is computed. Then, in the maximization step, the estimates of $(c_j)_{j=1,\dots,r}$ and $(\mathbf{D}_j[s])_{s=1,\dots,p_j}$ for $j = 1,\dots,r$ are updated using the conditional expectation of $(\mathbf{z}[t])_{t\in\mathbb{Z}}$. If $(\mathbf{\Sigma}_j)_{j=1,\dots,r}$ is treated as an unknown parameter, then this estimate is also updated during the maximization step.

In signal and image processing, researchers have found many successful applications for a class of bilinear models known as dictionary learning and non-negative matrix factorization. Dictionary learning, or sparse coding, follows from experimental work encoding natural images [15], [16]. In dictionary learning, we are given realizations $(\mathbf{x}_i)_{i=1,\dots,n}$ for which we want to find a dictionary in which the signals are appropriately sparse, *i.e.*

$$\mathbf{x}_i = \sum_{j=1}^{r} c_{i,j}\mathbf{d}_j + \mathbf{n}_i, \tag{4}$$

where $\mathbf{c}_i = (c_{i,j})_{j=1,\dots,r}$ are the dictionary coefficients for the i realization, $(\mathbf{d}_j)_{j=1,\dots,r}$ are the dictionary atoms, and $(\mathbf{n}_i)_{i=1,\dots,n}$ is random noise [17]. In this model, sparsity implies that $\|\mathbf{c}_i\|_0 \ll r$ for all $i = 1,\dots,n$. Non-negative matrix factorization adopts the generative model of dictionary learning (4), but the coefficients $\mathbf{c}_i$ need no longer be sparse, rather the coefficients and basis vectors $(\mathbf{d}_j)_{j=1,\dots,r}$ are constrained to be in the positive orthant [18].

Under a Gaussian noise model, the maximum likelihood estimator (MLE) for (4) is given by

$$\underset{\substack{\mathbf{D}\in\mathbb{R}^{d\times r}\\ \mathbf{c}_i\in\mathbb{R}^r,\ i=1,\dots,n}}{\arg\min}\quad \frac{1}{2n}\sum_{i=1}^{n}\|\mathbf{x}_i - \mathbf{D}\mathbf{c}_i\|^2, \tag{5}$$

where $\mathbf{D} = [\mathbf{d}_1|\cdots|\mathbf{d}_r]$. Equation (5) is non-convex and difficult to solve globally. Moreover, the parameters are non-identifiable since for any $\left\{\mathbf{D}, (\mathbf{c}_i)_{i=1,\dots,n}\right\}$ which minimizes (5), so too does $\left\{\mathbf{D}\mathbf{U}, \left(\mathbf{U}^{-1}\mathbf{c}_i\right)_{i=1,\dots,n}\right\}$ for any $\mathbf{U}\in\mathbb{R}^{r\times r}$ invertible. Numerous extensions to (5) and associated algorithms have been proposed for its solution. Some notable algorithms include the method of optimal directions [19], FOCUSS [20], k-SVD [17], feature-sign search algorithm [21], ER-SpUD [22], and sum-of-squares [23]. Non-negative matrix factorization has an expansive and independent literature from dictionary learning as it developed largely in parallel. Berry, *et al.* provide a good review of algorithms for non-negative matrix factorization [24].

Other generative signal models which feature a decomposition include principal component analysis (PCA) and independent component analysis (ICA). These models posit a linear decomposition of the signal into components as in (4). As contrasted with dictionary learning, PCA looks for orthogonal components which maximally account for variance in the data [25], while dictionary learning specifically allows for non-orthogonal components (atoms). ICA instead finds components for which the sources (coefficients) are statistically independent [26], wherein dictionary learning seeks sparseness in the coefficients. The autoregressive structure of ALM distinguishes it from dictionary learning, PCA, and ICA models, in which the realizations arise instantaneously (or causally) from the linear combination of unobserved sources. In ALM, a realization arises from the instantaneous linear combination of its own history as filtered through different autoregressive coefficients.

B. Notation

We denote vectors as boldface, lower-case letters and matrices as boldface, upper-case letters. The transpose of a real matrix $\mathbf{A}$ is denoted $\mathbf{A}^T$, and the complex conjugate of a complex matrix $\mathbf{A}$ is denoted $\mathbf{A}^*$. We denote the realization

of a random vector or matrix with a subscript, *i.e.* $\mathbf{x}_i$ or $\mathbf{A}_i$. We denote the element of a vector or matrix with a subscript without boldface, *i.e.* x_j or $A_{i,j}$. Inner products are denoted $\langle \cdot, \cdot \rangle$. We reference the Frobenius inner product of real matrices $\mathbf{A}, \mathbf{B}$, $\langle \mathbf{A}, \mathbf{B} \rangle_F = \operatorname{tr}\left(\mathbf{A}\mathbf{B}^T\right)$, where $\operatorname{tr}(\cdot)$ denotes the trace of a matrix. Norms are denoted with double bars. We use $\|\cdot\|$ to denote the vector 2-norm or matrix operator norm, *i.e.* largest singular value. $\|\cdot\|_F$ refers to the Frobenius norm, *i.e.* $\|A\|_F^2 = \langle \mathbf{A}, \mathbf{A} \rangle_F$. We denote the column-wise vectorization of a matrix $\operatorname{vec}(\cdot)$. We denote the Kronecker product as $\otimes$: for any $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{B}^{p \times q}$, $A \otimes B_{p(r-1)+v, q(s-1)+w} = A_{r,s} B_{v,w}$. For self-adjoint matrices $\mathbf{A}, \mathbf{B}$, $\mathbf{A} \succeq \mathbf{B}$ indicates that $\mathbf{A} - \mathbf{B}$ is non-negative; $\mathbf{A} \succ \mathbf{B}$ indicates that $\mathbf{A} - \mathbf{B}$ is positive. We use $\mathbb{N}$, $\mathbb{Z}$, and $\mathbb{R}$ to denote the set of natural numbers, integers, and real numbers respectively. We will denote probability as P and expectation as E. We use $\hat{\cdot}$ notation as shorthand for the Fourier transform on $\mathbb{Z}$, *e.g.* for $(z[t])_{t \in \mathbb{Z}}$, $\hat{z}(\omega) = \sum_{t \in \mathbb{Z}} e^{2\pi i \omega t} z[t]$. Realizations are assumed to be in $\mathbb{R}^d$ throughout. We use big O notation to relate the order of functions asymptotically. For instance, for real-valued functions f and g, $f(x) \sim \mathcal{O}(g(x))$ implies that f grows no faster than g as $x \to \infty$.

III. Methodology

A. Model

We can understand the ALM model as simultaneously the linearization of the MVAR model and a dictionary learning model for vector autoregressive processes. Let $(\mathbf{x}[t])_{t \in \mathbb{Z}}$, $(\mathcal{F}_t)_{t \in \mathbb{Z}}$, and $\mathbf{\Phi}$ be defined as in (3). Then, we can write (2) as

$$\begin{aligned} &\mathrm{P}\left(x_1[t] \leq x_1, \ldots, x_d[t] \leq x_d \mid \mathbf{c}, \mathcal{F}_{t-1}\right) \\ &= \mathbf{\Phi}\left(\mathbf{\Sigma}^{-1/2}\left(\mathbf{x} - \sum_{j=1}^{r} c_j \sum_{s=1}^{p} \mathbf{D}_j[s]\mathbf{x}[t-s]\right)\right) \end{aligned} \tag{6}$$

for any $\mathbf{x} \in \mathbb{R}^d$. In comparison to (3), we adopt the dictionary learning convention and do not constrain the mixing coefficients to be the parameters of a r-dimensional categorical random variable. Rather, we treat the mixing coefficients as random variables in a hierarchical model. For simplicity, we assume that all autoregressive components share a common model order, *i.e.* $p_j = p_{j'}$ for all $j, j' = 1, \ldots, r$, and that there is a single noise process with covariance $\Sigma \succeq \mathbf{0}$. These assumptions can be relaxed. We refer to $(\mathbf{D}_j[s])_{s=1,\ldots,p}$ as the j autoregressive component and $\mathbf{c}$ as the mixing coefficients of the $\mathrm{ALM}(p, r)$ process.

The autoregressive coefficients of a standard vector autoregressive model specify the causal mechanism by which variates in a multivariate signal interact. In this way, each autoregressive component $(\mathbf{D}_j[s])_{s=1,\ldots,p}$ of (2) can be interpreted as a separate network process. In fact, each component $(\mathbf{D}_j[s])_{s=1,\ldots,p}$ defines an infinite impulse response (IIR) filter with transfer function

$$\mathbf{H}_j(\omega) = \left(\mathbf{I} - \hat{\mathbf{D}}_j(\omega)\right)^{-1} \tag{7}$$

for $\omega \in [0, 1]$ and $\hat{\mathbf{D}}_j(\omega) = \sum_{s=1}^{p} e^{2\pi i \omega s} \mathbf{D}_j[s]$. The mixing coefficients describe how these network processes combine to bring about the observed signals.

VAR models are already notable for the challenges of high-dimensional model-fitting. A single realization of an $\mathrm{ALM}(p, r)$ process has more parameters than a single realization of a VAR(p) process, $rpd^2 + r$ versus pd^2. However, the benefit in terms of model complexity arises in the case of multiple realizations. Suppose that we observe $n > 1$ time-series realizations. If we attempt to fit a VAR(p) model to each realization, then we must estimate npd^2 parameters. For the $\mathrm{ALM}(p, r)$ model, by assuming that the realizations share a common set of autoregressive components but are differentiated by the mixing coefficients, we must estimate $rpd^2 + nr$ parameters. Only the mixing coefficients are linear in the number of realizations. Asymptotically, for fixed d, p, r, the relative model complexity of ALM to VAR depends on the ratio of the number of autoregressive components r to the complexity of a VAR(p) model pd^2:

$$\lim_{n \to \infty} \frac{rpd^2 + nr}{npd^2} = \frac{r}{pd^2}.$$

For analysis in the finite case, let $r < pd^2$. Then, the model complexity of VAR(p) exceeds that of $\mathrm{ALM}(p, r)$ for $n > (rpd^2)/(pd^2 - r)$.

Although the ALM model linearly combines the autoregressive components in (2), the components combine nonlinearly to generate the signal from white noise. This can be seen in the frequency domain of the moving average representation of (2),

$$\hat{\mathbf{x}}(\omega) = \mathbf{H}(\omega) \cdot \hat{\mathbf{n}}(\omega), \tag{8}$$

where $\mathbf{H}$ denotes the combined transfer function,

$$\mathbf{H}(\omega) = \left(\mathbf{I} - \sum_{j=1}^{r} c_j \hat{\mathbf{D}}_j(\omega)\right)^{-1}. \tag{9}$$

In general, we cannot relate the combined transfer function (9) to the component transfer functions (7), except in the following special cases addressed in Proposition 1.

Proposition 1. *Suppose that $(\mathbf{x}[t])_{t \in \mathbb{Z}}$ is a realization of an $\mathrm{ALM}(p, r)$ process with autoregressive components $\left\{(\mathbf{D}_j[s])_{s=1,\ldots,p} : j = 1, \ldots, r\right\}$ and mixing coefficients $(c_j)_{j=1,\ldots,r}$ such that $c_j \geq 0$ for all $j = 1, \ldots, r$ and $\sum_{j=1}^{r} c_j = 1$. For each $\omega \in [0, 1]$:*

1) *if for all $j = 1, \ldots, r$, $\hat{\mathbf{D}}_j(\omega)$ is self-adjoint and $\hat{\mathbf{D}}_j(\omega) \prec \mathbf{I}$, then*

$$\|\mathbf{H}(\omega)\| \leq \sum_{j=1}^{r} c_j \|\mathbf{H}_j(\omega)\|;$$

2) *if $R := \max_{j \in \{1,\ldots,r\}} \left\|\hat{\mathbf{D}}_j(\omega)\right\| < 1$, then for $\varphi(R) = \frac{1+R}{1-R}$,*

$$\|\mathbf{H}(\omega)\| \leq \varphi(R) \sum_{j=1}^{r} c_j \|\mathbf{H}_j(\omega)\|.$$

If $\left\{\hat{\mathbf{D}}_j(\omega) : j = 1, \ldots, r\right\}$ are not self-adjoint, one cannot expect an upper bound such as in (1) of Proposition 1. In (2) of Proposition 1, we achieve a similar result with the inclusion of a constant which grows exponentially as $R \to 1$. Ideally, we would like to replace $\varphi(R)$ with a constant independent of R. However, although the constant $\frac{1+R}{1-R}$ may not be optimal in (2) of Proposition 1, the following counterexample shows that in the worst case, the constant must depend on R.

Example. Consider the case $r = 2$, $c_1 = c_2 = \frac{1}{2}$, $\hat{\mathbf{D}}_1(\omega) = R\exp(i\arccos(R))$, $\hat{\mathbf{D}}_2(\omega) = R\exp(-i\arccos(R))$ then

$$\frac{\|\mathbf{H}(\omega)\|}{\sum_{j=1}^{r} c_j \|\mathbf{H}_j(\omega)\|} = \frac{1}{\sqrt{1-R^2}}.$$

Equation (8) provides insight into the stability and stationarity of the ALM (p, r) process. Stability implies the transfer of energy from noise to the signal decays sufficiently fast in time. This dampening effect preserves the first and second moments of the time-series. Accordingly, the stability of a vector autoregressive process implies wide-sense stationarity [1, Prop. 2.1]. We provide a sufficient condition for stability and wide-sense stationarity of an ALM (p, r) process in Proposition 2.

Proposition 2. *Let $(\mathbf{x}[t])_{t\in\mathbb{Z}}$ be a realization of an ALM (p, r) process with autoregressive components $\left\{(\mathbf{D}_j[s])_{s=1,\ldots,p} : j = 1, \ldots, r\right\}$ and mixing coefficients $(c_j)_{j=1,\ldots,r}$. Then, $(\mathbf{x}[t])_{t\in\mathbb{Z}}$ is stable if*

$$det\left(\mathbf{I} - \sum_{j=1}^{r} c_j \left(\sum_{s=1}^{p} \mathbf{D}_j[s] z^s\right)\right) \neq 0$$

for all $z \in \{z \in \mathbb{C} : |z| \leq 1\}$. If the process is stable, it is also wide-sense stationary.

Stability yields a well-defined autocovariance function,

$$\mathbf{\Gamma}[s] = \mathrm{E}\mathbf{x}[t]\mathbf{x}^T[t-s], \tag{10}$$

for $s \in \mathbb{Z}$. The autocovariance function corresponds with the spectral density function via a Fourier transform [27],

$$\hat{\mathbf{\Gamma}}(\omega) = \sum_{s\in\mathbb{Z}} e^{2\pi i\omega s}\mathbf{\Gamma}[s]. \tag{11}$$

We can use (8) and (11) to derive the autocovariance of an ALM (p, r) process. This result is given in Proposition 3.

Proposition 3. *Let $\left\{(\mathbf{D}_j[s])_{s=1,\ldots,p} : j = 1, \ldots, r\right\}$, $(c_j)_{j=1,\ldots,r}$, and noise covariance $\mathbf{\Sigma} \succeq \mathbf{0}$ define a stable realization of an ALM (p, r) process. The spectral density function is given by*

$$\hat{\mathbf{\Gamma}}(\omega) = \mathbf{H}(\omega)\mathbf{\Sigma}^{-1}\mathbf{H}^*(\omega),$$

where $\mathbf{H}(\cdot)$ is defined as in (9)*, and the autocovariance function is given by*

$$\mathbf{\Gamma}[s] = \int_0^1 e^{-2\pi i\omega s}\hat{\mathbf{\Gamma}}(\omega)d\omega.$$

B. Estimation

In this section, we address parameter estimation for the ALM model based on maximizing the likelihood of the realizations. From the likelihood model, we derive a non-convex optimization problem. We present a proximal gradient algorithm which guarantees convergence to a stationary point of the objective.

1) Likelihood of realizations: Suppose we observe a finite length realization of an ALM (p, r) process, $(\mathbf{x}[t])_{t=1,\ldots,m+p}$. As we can whiten the realizations, we can assume $\mathbf{n}[t] \sim_{\text{iid}} \mathcal{N}(\mathbf{0}, \mathbf{I})$ without loss of generality. If we take the first p observations $(\mathbf{x}[t])_{t=1,\ldots,p}$ as fixed and known, the negative log likelihood of the remaining realization is given by

$$-\log \mathrm{P}(\mathbf{x}[p+1], \ldots, \mathbf{x}[m+p] \mid \mathbf{D}_1, \ldots, \mathbf{D}_r, \mathbf{c}) = \frac{md}{2}\log(2\pi) + \sum_{t=p+1}^{m} \frac{1}{2}\left\|\mathbf{x}[t] - \sum_{s=1}^{p}\sum_{j=1}^{r} c_j \mathbf{D}_j[s]\mathbf{x}[t-s]\right\|^2. \tag{12}$$

Let us define $\mathbf{Y} = [\mathbf{x}[p+1]\cdots\mathbf{x}[m+p]]^T$, $\mathbf{D}_j = [\mathbf{D}_j[1]\cdots\mathbf{D}_j[p]]^T$, $\mathbf{c} = [c_1\cdots c_r]^T$, and

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}^T[p] & \cdots & \mathbf{x}^T[1] \\ \ddots & \ddots & \ddots \\ \mathbf{x}^T[m-1] & \cdots & \mathbf{x}^T[m-p] \end{bmatrix}.$$

Then, we can write (12) in stacked form,

$$-\log \mathrm{P}(\mathbf{Y}, \mathbf{X} \mid \mathbf{D}_1, \ldots, \mathbf{D}_r, \mathbf{c}) = \frac{md}{2}\log(2\pi) + \frac{1}{2}\left\|\mathbf{Y} - \mathbf{X}\begin{bmatrix}\mathbf{D}_1 & \cdots & \mathbf{D}_r\end{bmatrix}(\mathbf{c}\otimes\mathbf{I})\right\|_F^2. \tag{13}$$

From (13), we note that the negative log likelihood yields a bi-quadratic function in $\mathbf{D} = [\mathbf{D}_1\cdots\mathbf{D}_r]$ and $\mathbf{c}$.

Equation (13) also reveals a problem of intrinsically lower dimension. The unknown parameters, $\mathbf{D}$ and $\mathbf{c}$, lie in the pre-image of $\mathbf{X}$, a tall matrix of rank no more than $\min(pd, m)$. We define the sample autocovariance function as

$$\mathbf{R}[s] = \frac{1}{m+p-s}\sum_{t=s+1}^{m+p} \mathbf{x}[t]\mathbf{x}^T[t-s]. \tag{14}$$

In Proposition 4, we show that the sample autocovariance function $(\mathbf{R}[s])_{s=0,\ldots,p}$ is a sufficient statistic for estimation of the autoregressive components and mixing coefficients. That is, in lieu of the complete realization (md values), we require only the sample autocovariance function ($(p+1)d^2$ values) to evaluate the likelihood.

Proposition 4. *Let $(\mathbf{x}[t])_{t=1,\ldots,m+p}$ be a finite length realization of an ALM (p, r) process with autoregressive components $\left\{(\mathbf{D}_j[s])_{s=1,\ldots,p} : j = 1, \ldots, r\right\}$ and mixing coefficients $(c_j)_{j=1,\ldots,r}$. Then, the sample autocovariance function, $(\mathbf{R}[s])_{s=0,\ldots,p}$ is a sufficient statistic for the autoregressive components and mixing coefficients.*

Note that $\left(\frac{1}{m}\mathbf{X}^T\mathbf{Y}\right)^T = \left[\mathbf{R}^T[1]\cdots\mathbf{R}^T[p]\right]$, and

$$\frac{1}{m}\mathbf{X}^T\mathbf{X} = \begin{bmatrix} \mathbf{R}[0] & \cdots & \mathbf{R}[p-1] \\ \vdots & \ddots & \vdots \\ \mathbf{R}[p-1] & \cdots & \mathbf{R}[0] \end{bmatrix}.$$

If we denote $\mathbf{R}_0 = \frac{1}{m}\mathbf{X}^T\mathbf{Y}$ and $\mathbf{R} = \frac{1}{m}\mathbf{X}^T\mathbf{X}$, then by (13) and Proposition 4,

$$\begin{aligned} &-\log \mathrm{P}\left(\mathbf{R}[0], \ldots, \mathbf{R}[p] \mid \mathbf{D}, \mathbf{c}\right) \\ &\propto \left\langle \mathbf{R}_0, \mathbf{D}\left(\mathbf{c} \otimes \mathbf{I}\right)\right\rangle_F + \frac{1}{2}\left\langle \mathbf{D}\left(\mathbf{c} \otimes \mathbf{I}\right), \mathbf{R}\mathbf{D}\left(\mathbf{c} \otimes \mathbf{I}\right)\right\rangle_F. \end{aligned} \tag{15}$$

We denote $L\left(\mathbf{D}_1, \ldots, \mathbf{D}_r, \mathbf{c}\right) = \left\langle \mathbf{R}_0, \mathbf{D}\left(\mathbf{c} \otimes \mathbf{I}\right)\right\rangle_F + \frac{1}{2}\left\langle \mathbf{D}\left(\mathbf{c} \otimes \mathbf{I}\right), \mathbf{R}\mathbf{D}\left(\mathbf{c} \otimes \mathbf{I}\right)\right\rangle_F$.

2) Likelihood-based estimator: In order to infer the mixing coefficients and autoregressive components of the ALM model, we require not just a single realization but realizations of different mixtures of the autoregressive components. With only one realization, there are infinitely many decompositions which can explain the realizations with the same likelihood. In dictionary learning, the number of realizations required to recover the underlying dictionary scales polynomially with the number of components [22], [23]. We should expect that we need a number and length of realizations that are polynomial in the number of components and model order for reliable recovery of autoregressive components in the ALM model.

Suppose that we observe n finite length realizations of an $\mathrm{ALM}\left(p, r\right)$ process, $\left\{\left(\mathbf{x}_i[t]\right)_{t=1,\ldots,m+p} : i = 1, \ldots, n\right\}$, from a common set of autoregressive components $\left\{\left(\mathbf{D}_j[s]\right)_{s=1,\ldots,p} : j = 1, \ldots, r\right\}$. That is,

$$\mathbf{x}_i[t] = \sum_{s=1}^{p}\sum_{j=1}^{r} c_{i,j}\mathbf{D}_j[s]\mathbf{x}_i[t-s] + \mathbf{n}_i[t]. \tag{16}$$

Without loss of generality, we assume that $\mathbf{n}_i[t] \sim_{\text{iid}} \mathcal{N}\left(\mathbf{0}, \mathbf{I}\right)$ for all $i = 1, \ldots, n$. From (15), we derive a maximum likelihood estimator (MLE),

$$\underset{\substack{\mathbf{D}_j \in \mathbb{R}^{pd\times d},\ j=1,\ldots,r \\ \mathbf{c}_i \in \mathbb{R}^r,\ i=1,\ldots,n}}{\arg\min} \frac{1}{n}\sum_{i=1}^{n} L_i\left(\mathbf{D}_1, \ldots, \mathbf{D}_r, \mathbf{c}_i\right). \tag{17}$$

Here, L_i denotes the likelihood term defined by the sample autocovariance function of realization i, $\left(\mathbf{R}_i[s]\right)_{s=0,\ldots,p}$. However, the estimator of (17) presents multiple problems. First, we must estimate $rpd^2 + nr$ parameters from realization of $n(p+1)d^2$ values. That convergence of the sample autocovariance function to the true autocovariance function depends on m complicates this analysis. Regardless, there is a non-trivial regime where the problem is not well-posed. Second, (17) is non-convex. It is convex in $\mathbf{D}_j$ for all $j = 1, \ldots, r$ and $\mathbf{c}_i$ for all $i = 1, \ldots, n$ respectively, but not jointly. Consequently, we do not have algorithms which can globally solve the estimation problem efficiently. Third, the parameters are non-identifiable. For any $\mathbf{D}$ and $\left(\mathbf{c}_i\right)_{i=1,\ldots,n}$ which minimize (17), so too do $\mathbf{D}\mathbf{U}$ and $\left(\mathbf{U}^{-1}\mathbf{c}_i\right)_{i=1,\ldots,n}$ for any $\mathbf{U} \in \mathbb{R}^{rd\times rd}$ invertible. Finally, the minimizers are unique only up to permutation.

We can address many of the problems outlined above with a maximum *a posteriori* (MAP) estimator,

$$\begin{aligned} &\underset{\substack{\mathbf{D}_j \in \mathbb{R}^{pd\times d},\ j=1,\ldots,r \\ \mathbf{c}_i \in \mathbb{R}^r,\ i=1,\ldots,n}}{\arg\min} \frac{1}{n}\sum_{i=1}^{n} L_i\left(\mathbf{D}_1, \ldots, \mathbf{D}_r, \mathbf{c}_i\right) \\ &+ \frac{1}{n}\sum_{i=1}^{n} \mu\left\|\mathbf{c}_i\right\|_1 \ \text{ s.t. } \ \left\|\mathbf{D}_j\right\|_F = 1,\ j = 1, \ldots, r, \end{aligned} \tag{18}$$

where $\mu \in \mathbb{R}$. We assume that vector autoregressive processes comprise a sparse mixture of unit-norm autoregressive components in order to make the problem better posed. The mixing coefficient constraints correspond to a Laplace prior distribution, and the constraint on the autoregressive components corresponds to a uniform prior distribution on the Frobenius unit sphere. The constraints on the mixing coefficients eliminate ambiguity from arbitrary unitary transformations. The constraints on the autoregressive components fix the scale ambiguity problem. This leaves only a sign and permutation ambiguity. Unfortunately, the objective is still non-convex, and the unit norm constraint on the autoregressive components makes the feasible set of the problem non-convex as well. These model assumptions are not more exacting than those of ICA [26] and dictionary learning [22], [28].

C. Algorithm

Non-convex optimization problems such as (18) are intractable in the general case, requiring a brute force search within the feasible set to be guaranteed to find the global minimizer. Nonetheless, some simple and computationally efficient algorithms yield surprising convergence guarantees for problems arising from particular statistical models. For an excellent review as it relates to matrix completion and factorization, see Chi, Lu, and Chen [29]. Here, we consider the proximal alternating linearized minimization (PALM) algorithm of Bolte, Sabach, and Teboulle [6]. The PALM algorithm exploits the block separation of the unit norm constraint and penalty terms of (18) and the smoothness of the negative log likelihood term from (15). Importantly, the PALM algorithm provides a local convergence guarantee under reasonable assumptions.

First, we note that we can write the unit norm constraint of (18) as a penalty term, $\sum_{j=1}^{r}\mathcal{X}_{\|\cdot\|_F=1}\left(\mathbf{D}_j\right)$, where $\mathcal{X}_{\|\cdot\|_F=1}\left(\cdot\right) : \mathbb{R}^{pd\times d} \to \mathbb{R}$ is the characteristic function of the unit Frobenius norm. That is, for any $\mathbf{D} \in \mathbb{R}^{pd\times d}$,

$$\mathcal{X}_{\|\cdot\|_F=1}\left(\mathbf{D}\right) = \begin{cases} 0 & \left\|\mathbf{D}\right\|_F = 1 \\ \infty & \text{o.w.} \end{cases}. \tag{19}$$

This gives the objective of (18) a form

$$H\left(\mathbf{D}_1, \ldots, \mathbf{D}_r, \mathbf{c}_1, \ldots, \mathbf{c}_n\right) + \sum_{j=1}^{r} f\left(\mathbf{D}_j\right) + \sum_{i=1}^{n} g\left(\mathbf{c}_i\right), \tag{20}$$

where $H = (1/n)\sum_{i=1}^{n} L_i$, $f = \mathcal{X}_{\|\cdot\|_F=1}\left(\cdot\right)$, and $g = (\mu/n)\left\|\cdot\right\|_1$. Importantly, H is a smooth function of all variables and the non-smooth terms f and g are block separable.

PALM is of a class of proximal gradient methods which make use of the proximal operator [30], [31]. The *proximal operator* of a function $f : X \to \mathbb{R}$, $X \subset \mathbb{R}^d$ is defined for any $x \in X$ [32]

$$\mathrm{prox}_f\left(x\right) = \underset{y\in X}{\arg\min}\, f(y) + \frac{1}{2}\left\|x - y\right\|^2.$$

It defines a possibly set-valued map, $\mathrm{prox}_f\left(\cdot\right) : X \to 2^X$. The proximal operator generalizes the projection operator,

Algorithm 1. Proximal Alternating Linearized Minimization (PALM)

Initialize $\mathbf{D}_1^{(0)}, \dots, \mathbf{D}_r^{(0)}, \mathbf{c}_1^{(0)}, \dots, \mathbf{c}_n^{(0)}$
for $k = 1, 2, \dots$ **do**
  for $j = 1, \dots, r$ **do**
    $\mathbf{D}_j^{(k)} = \mathcal{P}\left(\mathbf{D}_j^{(k-1)} - \alpha_j^{(k)} \nabla_{\mathbf{D}_j} H\right)$
  for $i = 1, \dots, n$ **do**
    $\mathbf{c}_i^{(k)} = \mathcal{S}_{(\mu/n)\beta_i^{(k)}}\left(\mathbf{c}_i^{(k-1)} - \beta_i^{(k)} \nabla_{\mathbf{c}_i} H\right)$

and for (19), the proximal operator can be implemented as a projection,

$$\mathcal{P}(\mathbf{D}) = \operatorname{prox}_{\mathcal{X}_{\|\cdot\|_F=1}(\cdot)}(\mathbf{D}) = \frac{\mathbf{D}}{\|\mathbf{D}\|_F}. \tag{21}$$

Even though $\left\{\mathbf{D} \in \mathbb{R}^{pd \times d} : \|\mathbf{D}\|_F = 1\right\}$ is a non-convex set, (21) projects onto it uniquely. The proximal operator of $\gamma \|\cdot\|_1$ is the element-wise shrinkage operator, *i.e.* $s_\gamma(x) = \operatorname{sgn}(x) \max(0, |x| - \gamma)$. We denote the vector shrinkage operator as $\mathcal{S}_\gamma$.

PALM is an iterative algorithm in which we cyclically update each block variable, $\mathbf{D}_1, \dots, \mathbf{D}_r, \mathbf{c}_1, \dots, \mathbf{c}_n$. An iteration comprises a proximal gradient step and proximal update for each block variable. That is, each parameter is updated with a gradient step of H before a proximal update with respect to f or g as appropriate. The PALM algorithm is shown in Algorithm 1. Note that the gradient is evaluated with the most current values of each parameter. For example, when updating $\mathbf{D}_j^{(k)}$, the gradient is evaluated at $\mathbf{D}_{j-1}^{(k)}$ and $\mathbf{D}_j^{(k-1)}$. In the supplementary material, Algorithm 1 is illustrated in greater detail.

In using the PALM algorithm, we inherit its theoretical guarantees. This ensures that our algorithm finds a critical point of (18). The result is global in nature: independent of the initialization, the algorithm will converge to a critical point of the objective. It does not guarantee that we find a global minimum since the critical point may be a saddle point or local minimum.

Proposition 5. *Let the sequence of iterates* $\left(\left\{\mathbf{D}_1^{(k)}, \dots, \mathbf{D}_r^{(k)}, \mathbf{c}_1^{(k)}, \dots, \mathbf{c}_n^{(k)}\right\}\right)_{k \geq 0}$ *be generated according to Algorithm 1 and* $\mathbf{R}_i$ *denote the sample autocovariance function of* $(\mathbf{x}_i[t])_{t=1,\dots,m+p}$ *as in* (15). *Assume that* $\|\mathbf{R}_i\| < \infty$ *for all* $i = 1, \dots, n$ *and* $c_{i,j}^{(k)} < \infty$ *for all* $i = 1, \dots, n$, $j = 1, \dots, r$, *and* $k \geq 0$. *Let*

1) $0 < \alpha_j^{(k)} < \left\| \frac{1}{n} \sum_{i=1}^n c_{i,j}^{(k-1)} \mathbf{R}_i \right\|^{-1}$ *and*
2) $0 < \beta_i^{(k)} < \left\| \frac{1}{n} \mathbf{G}_i \right\|^{-1}$

for all $i = 1, \dots, n$ *and* $j = 1, \dots, r$, *where* $[G_i]_{j,j'} = \left| \left\langle \mathbf{D}_j^{(k)}, \mathbf{R}_i \mathbf{D}_{j'}^{(k)} \right\rangle_F \right|$. *Then, the the iterates will converge to a critical point of* (20).

As the ALM (p, r) model requires the estimation of many parameters, we must also consider the computational complexity of model fitting. We compare to the computational complexity of the least-squares estimator for fitting a VAR(p) model to each realization. Although the PALM algorithm is iterative, we incur a one-time cost of $\mathcal{O}\left(nmp^2d^2\right)$ in computing the sample autocovariance for each realization $\left\{\mathbf{R}_i = \mathbf{X}_i^T \mathbf{X}_i : i = 1, \dots, n\right\}$. For the regime of large m ($m > pd^2$), this computation dominates the complexity of the least-squares estimator [1, Eq. 3.2.7]. In the iterative phase of PALM, for large n ($n > r$), the complexity is dominated by multiplying the sample autocovariance and the autoregressive component for every combination of realization and component $\{\mathbf{R}_i \mathbf{D}_j : i = 1, \dots, n,\ j = 1, \dots, r\}$, a total cost of $\mathcal{O}\left(nrp^2d^3\right)$. Thus, we eliminate the dependence on the length of realization m in the iterative phase of PALM. Following estimation of the sample autocovariance for each realization, the least-squares estimator for the VAR(p) process requires a linear solve for each realization at a total cost of $\mathcal{O}\left(np^3d^3\right)$. Therefore, fitting an ALM (p, r) model with the PALM algorithm has comparable computational complexity to fitting a VAR(p) model to each realization for $r \sim \mathcal{O}(p)$.

D. Recoverability and conditioning

We would like to provide a set of algebraic and statistical conditions on the autoregressive components and mixing coefficients which would make (20) well-posed. In ICA, recoverability can be conditioned on the linear independence of the mixing vectors and a statistical condition on the coefficients [26, Def. 1]. Similarly, in dictionary learning, it is assumed that the dictionary components are either linearly independent in the undercomplete case [22] or incoherent in the overcomplete case [28]. Assumptions on the coefficients serve to control the correlation.

Naively, we might want to impose a condition on the incoherence of the autoregressive components, *i.e.*

$$\max_{j,j' \in \{1,\dots,r\}: j \neq j'} \langle \mathbf{D}_j, \mathbf{D}_{j'} \rangle \leq \tau \tag{22}$$

for some appropriate $\tau > 0$. However, recovery is not done with the gram matrix with j, j'-entries $\langle \mathbf{D}_j, \mathbf{D}_{j'} \rangle$. The coefficient recovery depends on the conditioning of the gram matrix $\mathbf{G}$ of Proposition 5. If we let $m \to \infty$, then we see that the autoregressive components must be incoherent in a geometry induced by the respective autocovariance functions. That is, we require

$$\max_{\substack{i \in 1,\dots,n \\ j,j' \in \{1,\dots,r\}: j \neq j'}} \langle \mathbf{D}_j, \mathbf{\Gamma}_i \mathbf{D}_{j'} \rangle \leq \tau \tag{23}$$

for some $\tau > 0$ to satisfy an incoherence condition. From Proposition 3 and (9), we observe that $\mathbf{\Gamma}_i$ depends nonlinearly on both the autoregressive components and mixing coefficients. Therefore, we cannot decouple the algebraic conditions on the autoregressive components from the statistical conditions on the coefficients.

IV. Simulations

In this section, we empirically characterize the performance of Algorithm 1 with simulated data. We generate the data according to the following model:

1) autoregressive coefficients $(\mathbf{D}_j)_{j=1,\dots,r}$ are drawn iid standard normal and then normalized via projection;

2) mixing coefficients $(\mathbf{c}_i)_{i=1,\dots,n}$ have support of size $s < r$; the support is chosen from a uniform distribution over sets of size s; then, the coefficient values are drawn from $\mathcal{N}\left(\mathbf{0}, r^{-1/2}\mathbf{I}\right)$; mixing coefficients are re-drawn until they satisfy Proposition 2; and
3) realizations $(\mathbf{x}_i[t])_{t=1,\dots,m}$ are generated causally using (2) with noise drawn from $\mathcal{N}\left(\mathbf{0}, d^{-1/2}\mathbf{I}\right)$; we discard an initial 2000 samples to allow for mixing.

For validation, we use $d = 5$, $r = 10$, $p = 2$, and $s = 3$ unless otherwise stated. The experiments in this section are implemented in Python using the ALM library available at github.com/addisonbohannon/alm.

In addition to evaluating estimators by negative log likelihood, we use a pseudo-metric for error in recovery of the autoregressive components that accounts for permutation and sign ambiguity. That is,

$$\mathrm{d}\left((\mathbf{D}_j)_{j=1,\dots,r}, (\mathbf{D}'_j)_{j=1,\dots,r}\right) = \underset{\mathbf{S}\in\mathcal{S},\mathbf{P}\in\Pi}{\arg\min} \left\| \begin{bmatrix} \mathrm{vec}\,(\mathbf{D}_1) \\ \vdots \\ \mathrm{vec}\,(\mathbf{D}_r) \end{bmatrix} - (\mathbf{S}\otimes\mathbf{I})(\mathbf{P}\otimes\mathbf{I}) \begin{bmatrix} \mathrm{vec}\,(\mathbf{D}'_1) \\ \vdots \\ \mathrm{vec}\,(\mathbf{D}'_r) \end{bmatrix} \right\|^2, \tag{24}$$

where $\mathcal{S}$ is the set of $r \times r$ diagonal matrices with entries ± 1, and Π is the set of $r \times r$ permutation matrices. When $(\mathbf{D}_j)_{j=1,\dots,r}$ and $(\mathbf{D}'_j)_{j=1,\dots,r}$ correspond to the fitted and generative autoregressive components respectively, we refer to (24) as the component error.

A. Performance

Since (18) is non-convex, we explore the component error of the estimated autoregressive components over random initializations of the algorithm. For the same set of realizations ($n = 1000$ and $m = 10\,000$), the algorithm begins with 5 distinct initializations. The initializations are selected using the same procedure as used for generating problems (step 1 above). The results are shown in Figure 1.

The most apparent observation in Figure 1 is the monotonic decrease in negative log likelihood. More significant is consistency of the negative log likelihood achieved by each of the respective initializations. Each distinct initialization achieves the same negative log likelihood to within $\mathcal{O}\left(10^{-5}\right)$. This would suggest that the algorithms are finding solutions in some symmetry set, but the variable performance in terms of component error complicates this interpretation. Apparently, there are many equally likely explanations for the realizations beyond that of the actual autoregressive components (and signed permutations) that generated the realizations. This may be a result of ill-posedness as discussed in Section III-B.

B. Effect of number and length of realizations

As described in Section III-B, we expect recovery to depend polynomially on model order and number of components. Here, we fix the model complexity and vary the scale of realizations with respect to number and length of realizations. Problem sizes are sampled in log-scale. We simulate 10 problems, vary the number and length of realizations available

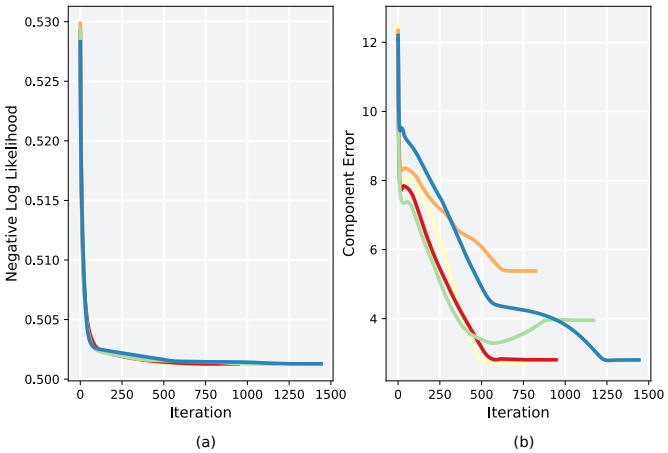


Fig. 1. Performance characterization of PALM algorithm. Each colored line corresponds to a unique initialization of the algorithm for the same set of realizations. (a) We show the negative log likelihood of the realizations (18) versus the iteration. We observe monotonically increasing likelihood (decreasing negative log likelihood) across all initializations. Moreover, all initializations reach a common likelihood. (b) We show the component error (24) against the iteration. The algorithm appears to have achieved the correct factorization for more than one initialization but not all initializations.

for fitting the ALM model, and for each problem, initialize the algorithm with 10 distinct initializations. We use the same initializations across all problem scales in order to understand how problem scale affects the estimation problem (18). We report the average (over the 10 problems) minimum (over the 10 initializations) component error and average (over the 10 problems) standard deviation (over the 10 initializations) in Figure 2.

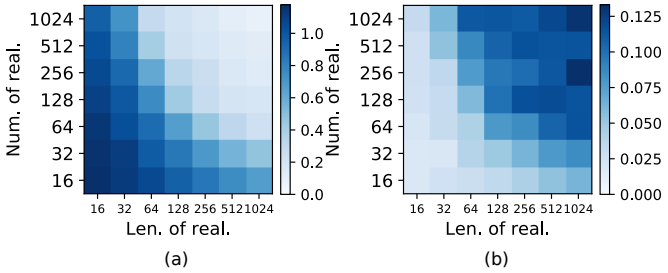


Fig. 2. Component error as a function of number and length of realizations (n and m respectively). (a) We show the component error (24) (normalized for the number of components) averaged over 10 problems sampled according to Section III-B. The per-sample component error used in the average is the minimum achieved from 10 distinct initializations. The component error decays faster in length of realization than number of realizations. (b) We show the standard deviations of component error among the 10 initializations averaged over the 10 problems. We observe that standard deviation increases with number and length of realizations.

Figure 2 provides insight into the surface of (18). Component error decays more rapidly with increasing length of realization than number of realizations. That the average component error achieved improves with additional realizations suggests that there exists a regime of n, m, p, and r for which the region of convergence to the true generative model increases. This would explain why for an equal number of initializations, the algorithm more reliably finds better minima. That standard deviation increases in this regime suggests that

the problem does not feature a benign landscape since some initializations achieve a component error far inferior to the global minimum.

We can gain additional insight into the results shown in Figure 2 by returning to (13). Suppose that we observe n realizations of length m from an ALM (p, r) process. Then, we can expand the negative log likelihood as follows for some $\left\{\mathbf{A}_i \in \mathbb{R}^{md \times d} : i = 1, \ldots, n\right\}$:

$$\begin{aligned}
&\sum_{i=1}^{n} \frac{1}{2} \left\| \mathbf{Y}_i - \mathbf{X}_i \mathbf{D} \left(\mathbf{c}_i \otimes \mathbf{I} \right) \right\|_F^2 \\
&= \sum_{i=1}^{n} \frac{1}{2} \left\| \mathbf{Y}_i - \mathbf{X}_i \mathbf{A}_i \right\|_F^2 \\
&- \sum_{i=1}^{n} \left\langle \mathbf{Y}_i - \mathbf{X}_i \mathbf{A}_i, \mathbf{X}_i \mathbf{A}_i - \mathbf{X}_i \mathbf{D} \left(\mathbf{c}_i \otimes \mathbf{I} \right) \right\rangle_F \\
&+ \sum_{i=1}^{n} \frac{1}{2} \left\| \mathbf{X}_i \mathbf{A}_i - \mathbf{X}_i \mathbf{D} \left(\mathbf{c}_i \otimes \mathbf{I} \right) \right\|_F^2 .
\end{aligned}$$

The first term on the right-hand side of the equality achieves a minimum at $\mathbf{A}_i = \left(\frac{1}{m} \mathbf{X}_i^T \mathbf{X}_i \right)^{-1} \left(\frac{1}{m} \mathbf{X}_i^T \mathbf{Y}_i \right)$ for all $i = 1, \ldots, n$. This is the least-squares estimator for vector autoregressive processes for each respective realization. Additionally, it eliminates the cross-product so that the only error contribution comes from the third term on the right-hand side. This term intuitively measures how well the autoregressive components decompose the least-squares estimator. Alternatively, the second and third terms on the right-hand side of the equality vanish for $\mathbf{A}_i = \mathbf{D} \left(\mathbf{c}_i \otimes \mathbf{I} \right)$ for all $i = 1, \ldots, n$. However, we have accepted sub-optimality in the first term since we can do no better than the per-iteration least-squares estimator. Regardless of how well the autoregressive components decompose the per-realization vector autoregressive models, there is an implicit dependence on the error of the least-squares estimator, which depends asymptotically on the length of realization m [1, Prop. 3.1].

As noted in Section III-A, in modeling multiple time-series realizations with an ALM (p, r) process instead of an individual VAR(p) process for each realization, we reduce the model complexity. This reduced model complexity should manifest in the component error. The error of the least-squares estimator for a VAR(p) model, to which we refer as coefficient error, has known asymptotic convergence rates of $1/\sqrt{m}$, where m is the length of the realization [1, Prop. 3.1]. When we independently estimate the coefficients of $n > 1$ realizations, the error will be additive. Therefore, we want to know at which point (if any), component error of n realizations of an ALM (p, r) process is as good as or better than $1/\sqrt{m}$. We do not derive error rates for ALM due to the complications discussed in Section III-D. However, we can consider small sample error numerically. The results of this analysis are shown in Figure 3.

In Figure 3, we compare the average error for the respective models using the PALM algorithm for fitting the ALM (p, r) model and the least-squares estimator for fitting the respective VAR(p) models. As expected, the average coefficient error for VAR decays logarithmically in m and independent of n. The component error of ALM decays more rapidly in m for large n and more slowly in m for small n. We observe an advantage to using the ALM (p, r) model for small m, but an advantage to using the VAR(p) model for small n. Notably, all of the simulated problems fall in the regime of greater model complexity for the n VAR(p) processes than that of the ALM (p, r) process, *i.e.* $n > (rpd^2)/(pd^2 - r)$.

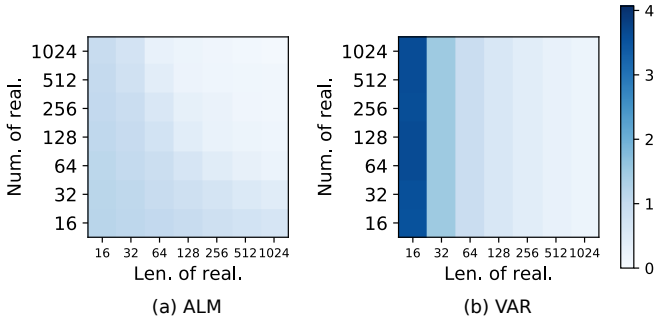
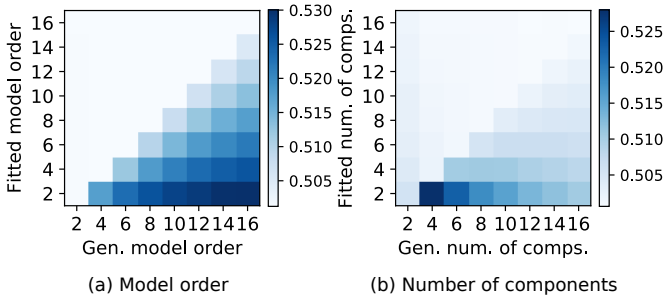


Fig. 3. Comparison of component error for ALM and coefficient error for VAR models. For comparison, error in (a) is normalized for number of components, and error in (b) is normalized for number of realizations. (a) The figure is the same as in Figure 2 (a). (b) We show the average coefficient error (normalized by the number of realizations) for 10 randomly sampled problems. As expected, the coefficient error of VAR(p) models decays logarithmically in the length of realization m, and this relationship is independent of the number of realizations n. We observe considerable advantage to using the ALM (p, r) model for small m, especially for large n. However, the component error decays much slower in the small n regime as compared to the coefficient error of VAR(p).

C. Model misspecification

We next investigate the robustness of the ALM model and PALM algorithm to model misspecification. Equation (2) offers an idealized generative model for signals. For signals observed in the real-world, we will not know the model order or number of autoregressive components. Therefore, we evaluate the performance of the ALM model when the data is generated from a different model complexity than that used to fit the data. We investigate misspecifying model order and number of autoregressive components separately. For a fixed number of autoregressive components, we generate data from ALM models of varying model orders ($n = 1000$, $m = 10\,000$), and we evaluate the fit of ALM models of varying model orders with respect to negative log likelihood. Similarly, for a fixed model order, we generate data from ALM models of varying number of autoregressive components, and we evaluate the fit of ALM models with varying number of autoregressive components. The results are shown in Figure 4.

In Figure 4(a), we observe a clear pattern of decreasing likelihood for increasing misspecification of the model order. When the fitted model order is less than the generative model order, we observe decreasing likelihood. Although the effect of misspecification increases with increasing generative model order, the relative effect of misspecification appears to decrease. That we use a fixed number and length of realizations for all problems likely accounts for this effect. These results suggest that standard model selection techniques such as Akaike information criterion (AIC) [33], [34] and Bayesian information criterion (BIC) [35] can be used to select



(a) Model order (b) Number of components

Fig. 4. Effect of model misspecification ($n = 1000$, $m = 10\,000$). (a) For a fixed number of autoregressive components, we show the negative log likelihood as we vary the generative and fitted model orders p. Increasing darkness of color corresponds to a decrease in likelihood. We observe a decrease in likelihood (increase in negative log likelihood) for underspecified models with the relative effect decreasing for increasing model order. (b) For a fixed model order, we show the negative log likelihood as we vary the generative and fitted number of components r. The same color scheme is used as in (a). We observe decreased likelihood for underspecified models, but the effect decreases with increasing number of components.

an appropriate model order. In light of the results in Figure 1, we cannot draw conclusions about the underlying component error.

In Figure 4(b), we observe again a decrease in likelihood for underspecified models. When the fitted number of components is less than the generative number of components, the likelihood decreases. The effect decays rapidly in the number of generative components. As in Figure 4(a), the effect likely results from the number and length of realization being fixed. The results indicate that AIC and BIC should successfully select an appropriate number of components. As mentioned above, these results do not provide insight into the underlying component error.

V. Application

In this section, we apply the ALM model to EEG recordings during sleep. In the following, we describe the data and how we fit the ALM model. We compare the use of the ALM model to that of MVAR and VAR models. We find that the features defined by fitting the ALM model better predict sleep stage. Moreover, the features form tight and separable clusters corresponding to sleep stage. This suggests that the autoregressive components learned by the ALM model are task relevant. We further investigate representative sleep stages according to the learned model and verify that it matches the sleep literature. Finally, we analyze the individual autoregressive components as network processes, finding distinct spectral and spatial features. The experiments in this section are implemented in Python using the ALM library available at github.com/addisonbohannon/alm.

A. Data

EEG signals, together with other physiological and pneumological data, allow practitioners to identify sleep stages in accordance with the standards defined by the American Academy of Sleep Medicine (AASM) [36]. The AASM guideline defines five different sleep stages: awake, N1 (drowsiness/transitional sleep), N2 (light sleep), N3 (deep sleep) and rapid eye movement (REM) sleep. The AASM guideline recommends assigning sleep stages based on 30 s non-overlapping intervals of recording. Sleep stages are identified based on the presence of unique oscillatory components along different EEG frequency bands, typically defined as δ (1-4 Hz), θ (4-8 Hz), α (8-13 Hz) and β (13-30 Hz). For example, N2 sleep is characterized by the presence of sleep spindles, which are high amplitude narrow-band oscillations in the 11-16 Hz frequency range, whereas N3 sleep is characterized by low-frequency high-amplitude waves in the 2-6 Hz frequency range (for more details see [36]).

The ISRUC-Sleep database [7] comprises 118 subjects across 3 different subgroups (Subgroups I, II and III). We focus our analysis on the subjects in subgroup III (10 subjects), those included as healthy controls. The EEG recordings include 6 channels (F4, C4, O2, F3, C3, O1) recorded at 200 Hz and referenced to electrodes placed on the left and right earlobes (*i.e.* $d = 6$). For each subject, we band-pass filter the full-length recording between 0.3 and 35 Hz using a Hamming windowed sinc FIR filter implemented in MNE-Python [37], following the recommendation in [7]. We extract non-overlapping, consecutive 30 s segments of data, together with their sleep stage labels, producing 800–1000 realizations per subject (*i.e.* $m = 6000$, $n = 800 - 1000$). Within realizations, we z-score normalize each channel. We summarize the preprocessing details in Figure 5. While sleep stage labels based on the AASM guideline are provided from two different experts, we use the labels from only the first expert in our preliminary analysis; Khalighi, *et al.* report minimal differences in the sleep stage labels between the two experts [7].

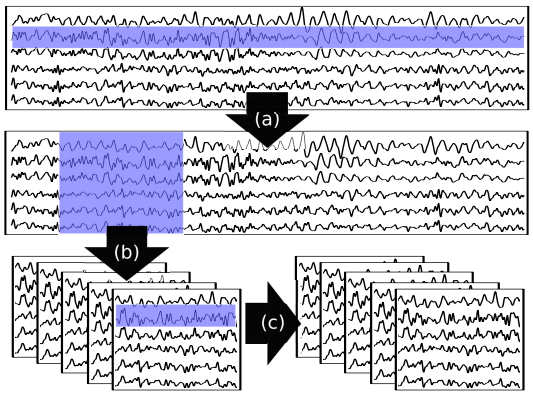


Fig. 5. Data processing procedure. For each subject: (a) we band-pass filter the full-length 200 Hz recording between 0.3 and 35 Hz using a Hamming windowed sinc FIR filter; (b) we partition the full-length recording into non-overlapping 30 s segments of data to yield n realizations of length m; and (c) within each realization, we z-score normalize the channels separately.

B. Model selection

Based on the model misspecification results from Section IV, we use a model selection criteria inspired by BIC to select an appropriate model complexity to fit the data [35]:

$$mBIC(p, r) = 2 * L(p, r) + \log(nm) \cdot \left(rpd^2 + nr\right). \quad (25)$$

In (25),

$$L(p,r) = \sum_{i=1}^{n} \frac{1}{2} \left\| \mathbf{Y}_i - \mathbf{X}_i \mathbf{D}^* \left(\mathbf{c}_i^* \otimes \mathbf{I} \right) \right\|_F^2 + \sum_{i=1}^{n} m\mu \left\| \mathbf{c}_i^* \right\|_1 , \tag{26}$$

where $\{\mathbf{c}_i^* : i = 1, \ldots, n\}$ and $\mathbf{D}^* = [\mathbf{D}_1^* \cdots \mathbf{D}_r^*]$ are the fitted mixing coefficients and autoregressive components from Algorithm 1. We believe that the issues with well-posedness discussed in Sections III-B and IV justify the harsh penalty of (25). We leave model selection for the ALM as a path for future inquiry.

We fit the ALM (p, r) model for $p \in \{4, 6, 8, 10, 12\}$ and $r \in \{2, 6, 10, 14, 18\}$. Due to the computational expense, we use subject 8 ($n = 999$) to select the best model. The minimum coincides with the ALM $(4, 10)$ model. We use this model for subsequent fitting and analysis of all subjects.

C. Reliability of autoregressive components

As in Section IV, the estimate that we obtain from the PALM algorithm depends on the initialization, achieving nearly identical penalized log likelihoods but providing different solutions. For the results shown, we select the initialization based on that which exhibits the spectral properties most consistent with those outlined in the AASM [36]. These results are included for comparison in the supplemental material. For ICA modeling of neuroimaging data, similar challenges are addressed with analysis of only components which arise reliably across trials and relate to task-relevant phenomena [38], [39]. Accordingly, data-driven analysis with ALM must be supported with domain expertise. Investigating the reliability of decomposition remains an open question for the ALM model.

D. Results

In comparing the ALM model to that of the MVAR and VAR models, we find that the features learned from fitting the ALM model better discriminate sleep stage, providing evidence for the appropriateness of the ALM model for sleep data. Additionally, we attempt to distill representations of each sleep stage from the learned model, and we highlight two fruitful techniques for analysis and interpretation of the learned autoregressive components. For most of the analysis, we report results for subject 8. The results for the remaining subjects are included in the supplementary material.

1) Comparison of ALM to MVAR and VAR models: We hypothesize that sleep stages are best modeled as instantaneous mixtures of network processes, making the ALM model more appropriate than MVAR or VAR. The fitted coefficients of the respective models define a feature map. We want to see if this feature map discriminates the sleep stages. If so, this would indicate that the models capture task-relevant phenomena. Our analysis suggests that the ALM model better describes the data than either the MVAR or VAR models.

For this analysis, we use the fitted mixing coefficients of the ALM $(4, 10)$ model as the features in a logistic regression model to predict the sleep stage. Then, we compare the results to that of MVAR and VAR. For each realization $(\mathbf{x}_i[t])_{t=1,\ldots,m}$, we use the mixing coefficients $\mathbf{c}_i \in \mathbb{R}^r$ as the feature vector. We evaluate the performance using a nested cross-validation with balanced classes. For each subject, the reported accuracy corresponds to the average multi-class accuracy across 5 folds for the best initialization within each fold according to a 5-fold cross-validation. For the MVAR model, we use $r = 10$ autoregressive components and $p = 4$ model order. We allow the mixture coefficients to vary for each realization, *i.e.* $\left\{ (c_{i,j})_{j=1,\ldots,r} : i = 1, \ldots, n \right\}$, to be consistent with the ALM framework. We estimate all parameters using an expectation-maximization algorithm detailed in the supplementary material. As for the ALM $(4, 10)$ model, we use the mixture coefficients as the features in the logistic regression and follow the same cross-validation procedure. For the VAR model, we use a model order of $p = 4$. We estimate the autoregressive coefficients for each realization using the least-squares estimator. We use the autoregressive coefficients as the features in a logistic regression with a five-fold cross validation with balanced classes. All analyses are performed using Sci-kit Learn [40]. Table I displays the results.

In Table I, we observe that all three models perform better than chance on each subject as expected performance for a random feature map is 20%. The ALM model has the greatest average accuracy as well as the best classification performance for 9 of the 10 subjects. We also observe that the ALM model achieves superior classification performance to the VAR model with many fewer parameters than VAR. For instance, let $n = 1000$. The ALM $(4, 10)$ model requires estimating $11\,440$ parameters. The VAR(4) model requires fitting $144\,000$ parameters.

We can also visualize the relationship of realizations using the respective feature map for the ALM, MVAR, and VAR models. For this, we use t-distributed stochastic neighbor embedding (t-SNE) [41], a non-linear projection technique. Figure 6 illustrates the results of the t-SNE analysis for each model on subject 8. We also plot an average feature $\bar{\mathbf{c}}_\ell$ for each sleep stage ℓ. Let $\mathcal{I}_\ell$ denote the set of realizations from sleep stage ℓ. Then, the sleep-stage centroid is given by

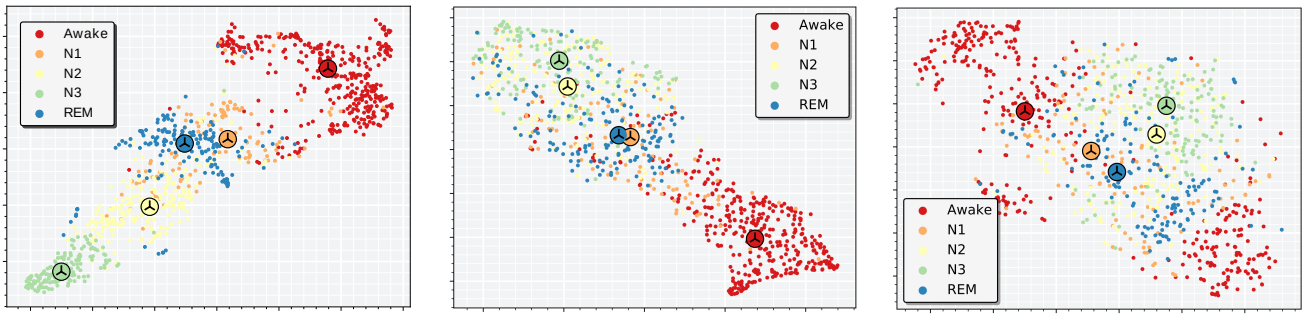
$$\bar{\mathbf{c}}_\ell = \frac{1}{|\mathcal{I}_\ell|} \sum_{i \in \mathcal{I}_\ell} \mathbf{c}_i \tag{27}$$

where $\mathbf{c}_i$ is the mixing coefficient or autoregressive coefficient corresponding to realization i.

In Figure 6, we observe that the ALM model features the best clustering. The sleep stages form tight and non-overlapping clusters. Interestingly, the relative positions appear to indicate relationships between the sleep stages: N1↔N2↔N3, N1↔Awake, and N1↔REM. The first two relationships are consistent with the AASM scoring manual, while the latter is not [36]. The MVAR model retains the same relationships between the sleep stage clusters, but the clusters are more diffuse and overlapping. The VAR model exhibits little clustering, with all classes overlapping. Figure 6 suggests that the mixture model captures something essential about the sleep data that cannot be modeled with the standard VAR model. This observation is consistent with observations in ICA modeling of EEG applications [42], [43].

TABLE I
SLEEP STAGE CLASSIFICATION RESULTS

Subject		1	2	3	4	5	6	7	8	9	10	Average
Accuracy (%)	ALM	71.5	64.6	71.7	71.0	76.8	63.1	70.2	80.5	67.7	56.0	**69.3**
	MVAR	43.1	33.2	47.5	44.1	49.2	40.4	58.2	62.8	34.5	47.8	**46.1**
	VAR	62.4	58.2	62.8	63.6	64.8	54.5	64.1	72.2	58.0	66.4	**62.7**



(a) ALM (b) MVAR (c) VAR

Fig. 6. Distribution of features for each realization of subject 8 for ALM, MVAR, and VAR models visualized using t-SNE [41]. Each colored dot represents the features for a 30 s realization of sleep data colored according to the expert-labeled sleep stage. Additionally, the centroids of the feature for each sleep stage (27) are plotted as large circles. (a) Projection of the ALM mixing coefficients. Sleep stages N1, N2, N3 and REM exhibit tight clustering whereas the awake class shows a broader distribution. The relative position of the realizations suggests a relationship between the sleep stages that is consistent with the AASM [36]. (b) Projection of the MVAR mixing coefficients. All sleep stages show some clustering with considerable overlap between N2–N3 and N1–REM. (c) Projection of the autoregressive coefficients of VAR. Sleep stages N1, N2, N3, and REM show some clustering, but all sleep stages overlap. Realizations of awake seemingly form three distinct clusters.

In comparing the ALM model to that of MVAR and VAR with a common model order and number of components, we directly address the hypothesis that an instantaneous mixture of network processes best describes the sleep data. The MVAR model has the capacity to fit the same autoregressive components, or network processes, as the ALM model. However, the MVAR will mix the autoregressive components nonlinearly. The VAR model has the capacity to exactly match the ALM model. However, the results from Table I and Figure 6 provide evidence that sleep stages are best modeled with ALM. Neither the additional model complexity of fitting nonlinear mixtures with MVAR nor the increased model capacity of VAR improve modeling performance.

2) Learned representation of sleep stages: Based on the results from the previous section, we use the mixing coefficient centroids of the ALM model to define an average transfer function for each sleep stage. With these transfer functions, we generate signals and periodograms to verify the learned representation.

We generate signals for each sleep stage ℓ with (2), the fitted autoregressive components $\left\{(\mathbf{D}_j[s])_{s=1,\dots,p} : j = 1,\dots,r\right\}$, and the average mixing coefficients defined in (27), $\left((\bar{\mathbf{c}}_\ell)_j\right)_{j=1,\dots,r}$, *i.e.*

$$\mathbf{x}_\ell[t] = \sum_{s=1}^{p}\sum_{j=1}^{r} (\bar{\mathbf{c}}_\ell)_j \, \mathbf{D}_j[s]\mathbf{x}_\ell[t-s] + \mathbf{n}[t]. \tag{28}$$

We also use the average mixing coefficents for each sleep stage ℓ to define a transfer function as in (9),

$$\mathbf{H}_\ell(\omega) = \left(\mathbf{I} - \sum_{j=1}^{r} (\bar{\mathbf{c}}_\ell)_j \, \hat{\mathbf{D}}_j(\omega)\right)^{-1}. \tag{29}$$

With these sleep stage filters, we generate a periodogram from the gain of the frequency response:

$$S_\ell(\omega) = \|\mathbf{H}_\ell(\omega)\|_2^2 . \tag{30}$$

The results of both the generated signal and periodogram for Subject 8 are shown in Figure 7. The results for the remaining subjects are included in the supplementary material.

In Figure 7, the awake periodogram shows a pronounced peak in the α band. Transitioning to N1, N2, and N3, we observe an increase in low frequency activity. The low frequency activity peaks in N3. This behavior broadly tracks that outlined in the AASM guidelines [36]. The spectral features of the N2, N3, and REM stages do not exhibit the same peaked activity as we observe in awake and N1. In the results for the remaining subjects (provided in the supplementary material), the peaked response of even the awake filter is not present. These results suggest that the average mixing coefficient insufficiently reflects the sleep stage, possibly as a result of the nonlinearity of the transfer function (see discussion in Section III-A and Prop. 1).

3) Analysis of the autoregressive components: We highlight the ability to analyze the individual autoregressive components of the fitted ALM model. We do so by first using the component IIR filters of (7) to analyze the spectral properties that they impart on signals and second performing functional

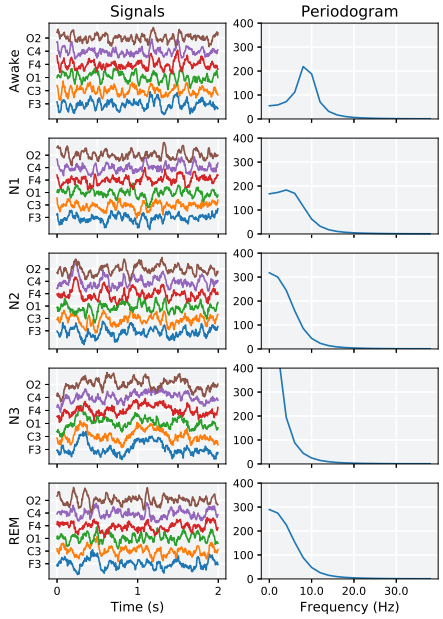


Fig. 7. Class-conditional signals and periodogram for subject 8. Each row corresponds to a sleep stage. For each sleep stage, we define an IIR filter using (29) and the centroid of the mixing coefficients from (27). In the left column, we generate signals from the stage-specific IIR filter acting on white noise. Each color corresponds to an EEG channel. In the right columns, periodograms show the spectral content of the sleep-stage filter. In the awake filter, we observe a peaked frequency response in the α band as expected. For the other filters, we observe increasing activity in δ and θ bands.

connectivity analysis of the channels. This allows us to understand the spectral and spatial properties of the autoregressive components.

To analyze the spectral properties of the autoregressive components, we generate a component-specific periodogram $S_j(\cdot) = \|\mathbf{H}_j(\cdot)\|_2^2$ where $\mathbf{H}_j$ is defined as in (7). The component periodograms for subject 8 are shown in Figure 8. Periodograms for the remaining subjects are included in the supplementary material.

Figure 8 shows two notable global features: the components partition the frequency domain and the gain decreases with increasing frequency. Through a succession of peaked responses, the components respond to every frequency from 0-35 Hz and exhibit a particular dense covering of 0-25 Hz: component 7 responds to 0-10 Hz; component 4 responds to 10-12 Hz; component 5 responds to 11-13 Hz; component 10 responds to 13-15 Hz; component 8 responds to 15-17 Hz; component 2 responds to 18-20 Hz. The dense covering of frequencies from 0-25 Hz covers the relevant frequencies for sleep activity as reported in the AASM scoring manual [36]. Interestingly, component 1 responds at 18-20 Hz and again at 30-32 Hz. Components 3, 6, and 9 apparently do not have peaked responses. The other notable phenomenon of Figure 8 is the decrease in gain for increasing frequency. This matches the so-called $1/f$ drop-off in power observed in EEG recordings [44].

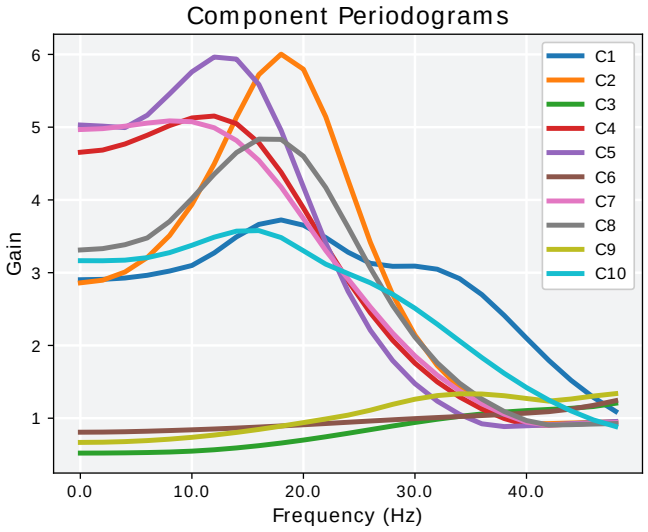


Fig. 8. Component periodograms for subject 8. Each color corresponds to a unique autoregressive component. For each autoregressive component, the gain of the frequency response (30) is plotted from 0-50 Hz. 7 of the 10 autoregressive components show peaked frequency responses. We observe that the autoregressive components partition the frequency domain and collectively display a decrease in cumulative gain for increasing frequency.

In addition to analysis of the frequency response of the individual autoregressive components, we can analyze the spatial relationships of the autoregressive components. Causal time-series modeling like that of autoregressive models captures functional connectivity [3]. Here, we use the directed transfer function (DTF), a multivariate version of Granger causality, to infer the functional connectivity between two EEG channels. The DTF between two channels d_1, d_2 is given by [45]

$$\gamma^2_{d_1,d_2}(\omega) = \frac{\left|[H_j]_{d_1,d_2}(\omega)\right|^2}{\sum_{d_2'=1}^{d}\left|[H_j]_{d_1,d_2'}(\omega)\right|^2}, \tag{31}$$

where $\mathbf{H}_j$ is the component transfer function of the j autoregressive component as defined in (7). The DTF provides a directed measure of connectivity among EEG channels, *i.e.* $\gamma_{d_1,d_2}(\omega)$ corresponds to the effect channel d_2 has on d_1 at frequency ω. We visualize the results of a functional connectivity analysis for each autoregressive component of subject 8 in Figure 9. The results for the remaining subjects are included in the supplementary material.

In Figure 9, we evaluate functional connectivity among the EEG channels within the EEG frequency bands: δ (1-4 Hz), θ (4-8 Hz), α (8-13 Hz) and β (13-30 Hz). We observe considerable variability in the connectivity patterns among the components. Component 8, which has a peaked frequency response, shows consistency in connectivity across all frequency bands. Component 6, which has no peaked frequency response, exhibits distinct connectivity patterns for each frequency band to include lateral connectivity in the frontal lobe in α-band and lateral connectivity in the occipital lobe in β-band. The spectral analysis of Figure 8 together with Figure 9 allows us to interrogate the unique contribution of each autoregressive component to the model.

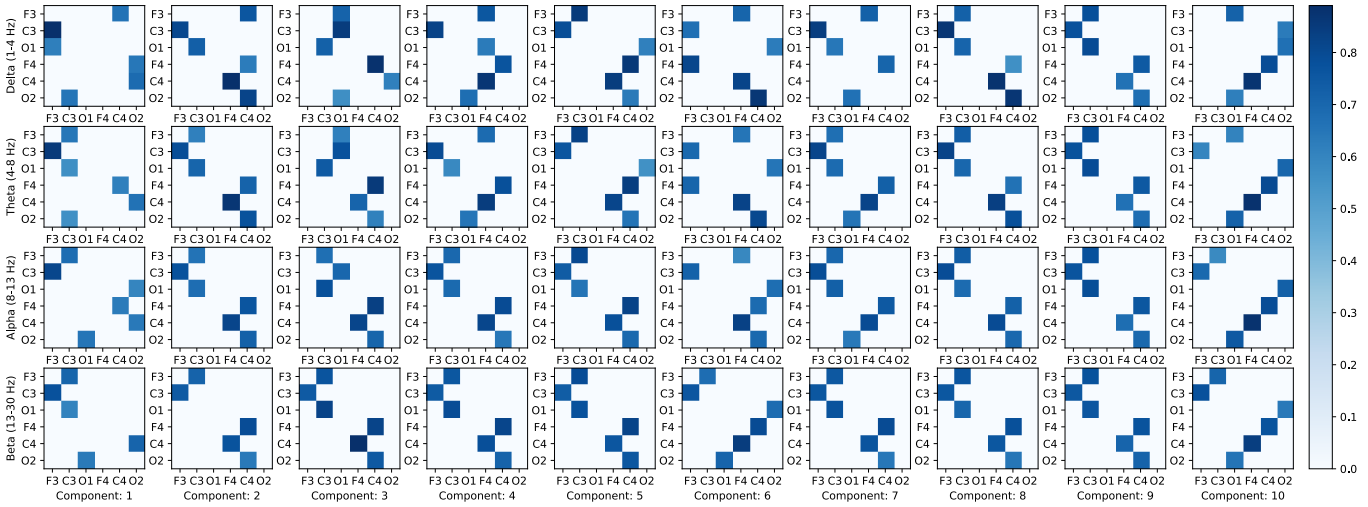


Fig. 9. Connectivity matrices for subject 8. Each column of figures corresponds to one autoregressive component, and each row corresponds to a frequency band. The d_1, d_2-element of each plot corresponds to the average DTF (31) for the frequency interval. Recall that this is a directed relationship, *i.e.* $d_2 \to d_1$. Only significant edges are shown in the plot ($p < 0.05$) [46]. The top-left and bottom-right quadrants of the connectivity plots correspond to intra-hemisphere connections. Some components such as 8 exhibit consistent connections across all frequency bands, while others such as 6 exhibit considerable variability between frequency bands. For instance, component 6 features lateral connectivity in the frontal lobe in α-band and lateral connectivity in the occipital lobe in β-band.

VI. Conclusion

In this work, we introduce a new model for analyzing time-series data which features the simultaneous mixture of network processes. We derive a likelihood-based estimator for the autoregressive components and mixing coefficients from multiple realizations of an ALM (p, r) process. Although the likelihood-based estimator yields a non-convex optimization problem, we show that the PALM algorithm performs capably. The validation results provide evidence that the region of convergence to the global minimum increases with increasing number and length of realizations. We fit the ALM model to EEG recordings of healthy individuals during sleep and show that ALM can disambiguate meaningful autoregressive components that correspond with network processes used for classifying sleep stages.

We identify two inter-related limitations of ALM in the application to real-world data: reliability of estimates across initializations and analytical understanding of what makes a well-posed problem. As discussed in Section III-B, the ALM model does not lend itself to separable conditions on the autoregressive components and mixing coefficients that facilitate recoverability. Practically, the well-posedness of the problem manifests as variability in estimates between different initializations, as discussed in Section V. These limitations require further investigation.

The ALM model provides numerous interesting directions of inquiry. First, it is worth noting that this work immediately generalizes to any proper, lower semi-continuous function g. Here, we consider a ℓ_1-norm penalty on the mixing coefficients, but we could as easily consider a ℓ_0-norm penalty or constrain the mixing coefficients to be non-negative or in the probability simplex. We would only need update the proximal function applied in the coefficient update of Algorithm 1. Next, we might want to know under what conditions the set of global minima of (18) coincide with the generating parameters as is shown for dictionary learning in [47]. Finally, we could consider estimating the autoregressive components using non-likelihood-based techniques such as generative adversarial networks [48].

We believe that the ALM model offers a powerful generative model for data-driven analysis of real-world time-series.

References

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

[2] B. S. Bernanke, J. Boivin, and P. Eliasz, "Measuring the effects of monetary policy: A factor-augmented vector autoregressive (favar) approach," *The Quarterly journal of economics*, vol. 120, no. 1, pp. 387–422, 2005.

[3] L. Harrison, W. D. Penny, and K. Friston, "Multivariate autoregressive modeling of fmri time series," *Neuroimage*, vol. 19, no. 4, pp. 1477–1491, 2003.

[4] K. J. Friston, L. Harrison, and W. Penny, "Dynamic causal modelling," *Neuroimage*, vol. 19, no. 4, pp. 1273–1302, 2003.

[5] K. J. Friston, "Functional and effective connectivity: A review," *Brain connectivity*, vol. 1, no. 1, pp. 13–36, 2011.

[6] J. Bolte, S. Sabach, and M. Teboulle, "Proximal alternating linearized minimization for nonconvex and nonsmooth problems," *Mathematical Programming*, vol. 146, no. 1-2, pp. 459–494, 2014.

[7] S. Khalighi, T. Sousa, J. M. Santos, and U. Nunes, "Isruc-sleep: A comprehensive public dataset for sleep researchers," *Computer methods and programs in biomedicine*, vol. 124, pp. 180–192, 2016.

[8] N. D. Le, R. D. Martin, and A. E. Raftery, "Modeling flat stretches, bursts outliers in time series using mixture transition distribution models," *Journal of the American Statistical Association*, vol. 91, no. 436, pp. 1504–1515, 1996.

[9] C. S. Wong and W. K. Li, "On a mixture autoregressive model," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 62, no. 1, pp. 95–115, 2000.

[10] P. W. Fong, W. K. Li, C. Yau, and C. Wong, "On a mixture vector autoregressive model," *Canadian Journal of Statistics*, vol. 35, no. 1, pp. 135–150, 2007.

[11] F. Bec, A. Rahbek, and N. Shephard, "The acr model: A multivariate dynamic mixture autoregression," *Oxford Bulletin of Economics and Statistics*, vol. 70, no. 5, pp. 583–618, 2008.

[12] M. J. Dueker, Z. Psaradakis, M. Sola, and F. Spagnolo, "Multivariate contemporaneous-threshold autoregressive models," *Journal of Econometrics*, vol. 160, no. 2, pp. 311–325, 2011.

[13] L. Kalliovirta, M. Meitz, and P. Saikkonen, "Gaussian mixture vector autoregression," *Journal of econometrics*, vol. 192, no. 2, pp. 485–498, 2016.

[14] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[15] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, p. 607, 1996.

[16] ——, "Sparse coding with an overcomplete basis set: A strategy employed by v1?" *Vision research*, vol. 37, no. 23, pp. 3311–3325, 1997.

[17] M. Aharon, M. Elad, and A. Bruckstein, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on signal processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[18] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, p. 788, 1999.

[19] K. Engan, S. O. Aase, and J. H. Husoy, "Method of optimal directions for frame design," in *Proceedings of the 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing*, IEEE, vol. 5, 1999, pp. 2443–2446.

[20] K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T.-W. Lee, and T. J. Sejnowski, "Dictionary learning algorithms for sparse representation," *Neural computation*, vol. 15, no. 2, pp. 349–396, 2003.

[21] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Advances in neural information processing systems*, 2007, pp. 801–808.

[22] D. A. Spielman, H. Wang, and J. Wright, "Exact recovery of sparsely-used dictionaries," in *Conference on Learning Theory*, 2012, pp. 37–1.

[23] B. Barak, J. A. Kelner, and D. Steurer, "Dictionary learning and tensor decomposition via the sum-of-squares method," in *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, ACM, 2015, pp. 143–151.

[24] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons, "Algorithms and applications for approximate nonnegative matrix factorization," *Computational statistics & data analysis*, vol. 52, no. 1, pp. 155–173, 2007.

[25] I. Jolliffe, *Principal component analysis*, 2nd. Springer, 2002.

[26] P. Comon, "Independent component analysis, a new concept?" *Signal processing*, vol. 36, no. 3, pp. 287–314, 1994.

[27] M. Priestley, *Spectral Analysis and Time Series*, ser. Probability and mathematical statistics. Academic Press, 1981, vol. 2nd.

[28] S. Arora, R. Ge, and A. Moitra, "New algorithms for learning incoherent and overcomplete dictionaries," in *Conference on Learning Theory*, 2014, pp. 779–806.

[29] Y. Chi, Y. M. Lu, and Y. Chen, "Nonconvex optimization meets low-rank matrix factorization: An overview," *IEEE Transactions on Signal Processing*, 2019.

[30] P. L. Combettes and V. R. Wajs, "Signal recovery by proximal forward-backward splitting," *Multiscale Modeling & Simulation*, vol. 4, no. 4, pp. 1168–1200, 2005.

[31] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics*, vol. 57, no. 11, pp. 1413–1457, 2004.

[32] N. Parikh, S. Boyd, *et al.*, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[33] H. Akaike, "Information theory and an extension of the maximum likelihood principle," in *2nd International Symposium on Information Theory*, Akademiai Kiado, 1973, pp. 267–281.

[34] H. Bozdogan, "Model selection and akaike's information criterion (aic): The general theory and its analytical extensions," *Psychometrika*, vol. 52, no. 3, pp. 345–370, 1987.

[35] G. Schwarz, "Estimating the dimension of a model," *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[36] R. B. Berry, R. Brooks, C. E. Gamaldo, S. M. Harding, C. Marcus, B. V. Vaughn, *et al.*, *The aasm manual for the scoring of sleep and associated events*, 2nd, American Academy of Sleep Medicine, 2012.

[37] A. Gramfort, M. Luessi, E. Larson, D. A. Engemann, D. Strohmeier, C. Brodbeck, R. Goj, M. Jas, T. Brooks, L. Parkkonen, *et al.*, "Meg and eeg data analysis with mne-python," *Frontiers in neuroscience*, vol. 7, p. 267, 2013.

[38] M. J. McKeown, S. Makeig, G. G. Brown, T.-P. Jung, S. S. Kindermann, A. J. Bell, and T. J. Sejnowski, "Analysis of fmri data by blind separation into inde-

pendent spatial components," *Human brain mapping*, vol. 6, no. 3, pp. 160–188, 1998.

[39] S. Makeig, M. Westerfield, T.-P. Jung, J. Covington, J. Townsend, T. J. Sejnowski, and E. Courchesne, "Functionally independent components of the late positive event-related potential during visual spatial attention," *Journal of Neuroscience*, vol. 19, no. 7, pp. 2665–2680, 1999.

[40] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[41] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[42] A. Delorme, J. Palmer, J. Onton, R. Oostenveld, and S. Makeig, "Independent eeg sources are dipolar," *PloS one*, vol. 7, no. 2, 2012.

[43] S.-H. Hsu, L. Pion-Tonachini, J. Palmer, M. Miyakoshi, S. Makeig, and T.-P. Jung, "Modeling brain dynamic state changes with adaptive mixture independent component analysis," *NeuroImage*, vol. 183, pp. 47–61, 2018.

[44] G. Buzsáki and A. Draguhn, "Neuronal oscillations in cortical networks," *Science*, vol. 304, no. 5679, pp. 1926–1929, 2004.

[45] M. Kamiński, M. Ding, W. A. Truccolo, and S. L. Bressler, "Evaluating causal relations in neural systems: Granger causality, directed transfer function and statistical assessment of significance," *Biological cybernetics*, vol. 85, no. 2, pp. 145–157, 2001.

[46] S. Maslov and K. Sneppen, "Specificity and stability in topology of protein networks," *Science*, vol. 296, no. 5569, pp. 910–913, 2002.

[47] R. Gribonval and K. Schnass, "Dictionary identification-sparse matrix-factorisation via ℓ_1-minimisation," *IEEE Transactions on Information Theory*, vol. 56, no. 7, pp. 3523–3539, 2010.

[48] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[49] H. V. Poor, *An introduction to signal detection and estimation*, 2nd. Springer-Verlag, 1994.

APPENDIX

Proof of Proposition 1. For simplicity of notation, we will omit the ω argument of the matrix-valued functions.

(1) Assume $\hat{\mathbf{D}}_j$ self-adjoint and $\hat{\mathbf{D}}_j \prec \mathbf{I}$ for all $j = 1, \dots, r$. Note

$$\|\mathbf{H}_j\| = \left\|(\mathbf{I} - \hat{\mathbf{D}}_j)^{-1}\right\| = \frac{1}{1 - \lambda_{\max}\left(\hat{\mathbf{D}}_j\right)},$$

where $\lambda_{\max}(\mathbf{A})$ denotes the largest eigenvalue of $\mathbf{A}$. A similar identity holds for $\mathbf{H} = (\mathbf{I} - \sum_{j=1}^r c_j \hat{\mathbf{D}}_j)^{-1}$. Let $\mathbf{v}$ be the normalized principal eigenvector of $\mathbf{D}_0 := \sum_{j=1}^r c_j \hat{\mathbf{D}}_j$. Thus, $\|\mathbf{H}\| = 1/(1 - \langle \mathbf{D}_0 \mathbf{v}, \mathbf{v}\rangle)$. Recall also that $\left\langle \hat{\mathbf{D}}_j \mathbf{v}, \mathbf{v} \right\rangle \leq \lambda_{\max}\left(\hat{\mathbf{D}}_j\right)$. After noting that $f(x) = 1/(1-x)$ is convex on $x < 1$, the statement follows from Jensen's inequality:

$$\begin{aligned}
\left\|(\mathbf{I} - \mathbf{D}_0)^{-1}\right\| &= \frac{1}{1 - \sum_{j=1}^r c_j \left\langle \hat{\mathbf{D}}_j \mathbf{v}, \mathbf{v} \right\rangle} \\
&\leq \sum_{j=1}^r c_j \frac{1}{1 - \left\langle \hat{\mathbf{D}}_j \mathbf{v}, \mathbf{v} \right\rangle} \\
&\leq \sum_{j=1}^r c_j \frac{1}{1 - \lambda_{\max}\left(\hat{\mathbf{D}}_j\right)} \\
&= \sum_{j=1}^r c_j \left\|\left(\mathbf{I} - \hat{\mathbf{D}}_j\right)^{-1}\right\|.
\end{aligned}$$

(2) Assume now that $\left\|\hat{\mathbf{D}}_j\right\| \leq R < 1$. Then $\mathbf{D}_0$ satisfies $\|\mathbf{D}_0\| \leq R$ as well. Note

$$\begin{aligned}
1 = \left\|(\mathbf{I} - \hat{\mathbf{D}}_j)(\mathbf{I} - \hat{\mathbf{D}}_j)^{-1}\right\| &\leq \left\|\mathbf{I} - \hat{\mathbf{D}}_j\right\| \left\|(\mathbf{I} - \hat{\mathbf{D}}_j)^{-1}\right\| \\
&\leq (1+R)\left\|(\mathbf{I} - \hat{\mathbf{D}}_j)^{-1}\right\|.
\end{aligned}$$

Then, $1 \leq (1+R)\sum_{j=1}^r c_j \left\|(\mathbf{I} - \hat{\mathbf{D}}_j)^{-1}\right\|$. On the other hand,

$$\left\|(\mathbf{I} - \mathbf{D}_0)^{-1}\right\| = \frac{1}{1 - \lambda_{\max}(\mathbf{D}_0)} \leq \frac{1}{1-R}.$$

We obtain the desired result by multiplying the last two inequalities. □

Proof of Proposition 2. A stable p-order autoregressive process with autoregressive coefficients $(\mathbf{A}[s])_{s=1,\dots,p}$ satisfies

$$\det\left(\mathbf{I} - \sum_{s=1}^p \mathbf{A}[s] z^p\right) \neq 0$$

for $|z| \leq 1$ [1, Eq. 2.1.12]. Let $\left(\sum_{j=1}^r c_j \mathbf{D}_j[s]\right)_{s=1,\dots,p}$ define an ALM (p, r). Then, the result follows from substitution of $\mathbf{A}[s] = \sum_{j=1}^r c_j \mathbf{D}_j[s]$. □

Proof of Proposition 3. Provided that the mixing coefficients are finite, the autocovariance function and spectral density function are in one-one correspondence since the components are square summable,

$$\sum_{s=1}^p \left\|\sum_{j=1}^r c_j \mathbf{D}_j[s]\right\|_F^2 < \infty.$$

Therefore, we can derive the autocovariance function as the Fourier transform of the spectral density function [27],

$$\hat{\mathbf{\Gamma}}(\omega) = \mathrm{E}\hat{\mathbf{x}}(\omega)\hat{\mathbf{x}}^*(\omega).$$

We can substitute (8) to get

$$\hat{\mathbf{\Gamma}}(\omega) = \mathbf{H}(\omega)\left(\mathrm{E}\hat{\mathbf{n}}(\omega)\hat{\mathbf{n}}^*(\omega)\right)\mathbf{H}(\omega)^*.$$

The middle term simplifies to $\mathbf{\Sigma}$. Then, by taking the Fourier transform, we recover the desired result. □

Proof of Proposition 4. We will use the factorization theorem, see *e.g.* [49, Prop. IV.C.1], which says that the sample autocovariance function $(\mathbf{R}[s])_{s=1,\dots,p}$ is a sufficient statistic for the autoregressive components and mixing coefficients if the likelihood of the realization $(\mathbf{x}[t])_{t=1,\dots,m+p}$ can be factored into the product of a function which depends on the sample autocovariance, autoregressive components, and mixing coefficients and a function which depends only on the realization. That is,

$$\begin{aligned}&\mathrm{P}\left((\mathbf{x}[t])_{t=p+1,\dots,m+p}\middle|\left\{(\mathbf{D}_j[s])_s : j\right\},(c_j)_j\right)\\&= f\left((\mathbf{x}[t])_{t=p+1,\dots,m+p}\right) g\left((\mathbf{R}[s])_s\middle|\left\{(\mathbf{D}_j[s])_s : j\right\},(c_j)_j\right)\end{aligned}$$

for some functions f and g. Without loss of generality, let us assume $\mathbf{n}[t] \sim_{\text{iid}} \mathcal{N}(\mathbf{0},\mathbf{I})$. The joint likelihood of the realization under the ALM (p,r) model is

$$\begin{aligned}&(2\pi)^{md/2}\,\mathrm{P}\left((\mathbf{x}[t])_{t=p+1,\dots,m+p}\right)\\&=\prod_{t=p+1}^{m+p}\exp\left(-\frac{1}{2}\left\|\mathbf{x}[t]-\sum_{s=1}^{p}\sum_{j=1}^{r}c_j\mathbf{D}_j[s]\mathbf{x}[t-s]\right\|^2\right)\\&=\exp\left(-\sum_{t=p+1}^{m+p}\frac{1}{2}\left\|\mathbf{x}[t]-\sum_{s=1}^{p}\sum_{j=1}^{r}c_j\mathbf{D}_j[s]\mathbf{x}[t-s]\right\|^2\right).\end{aligned}$$

Here, we have conditioned on the first p observations of each time-series. By expanding the norm and distributing the summation in the exponential, we derive our desired sample autocovariance terms:

$$\begin{aligned}&-\sum_{t=p+1}^{m+p}\frac{1}{2}\left\|\mathbf{x}[t]-\sum_{s=1}^{p}\sum_{j=1}^{r}c_j\mathbf{D}_j[s]\mathbf{x}[t-s]\right\|^2\\&=-\sum_{t=p+1}^{m+p}\frac{1}{2}\left\|\mathbf{x}[t]\right\|^2+m\sum_{s=1}^{p}\sum_{j=1}^{r}c_j\left\langle\mathbf{R}[s],\mathbf{D}_j[s]\right\rangle_F\\&-\frac{m}{2}\sum_{s,s'=1}^{p}\sum_{j,j'=1}^{r}c_jc_{j'}\left\langle\mathbf{D}_j[s],\mathbf{D}_{j'}[s']\mathbf{R}[s-s']\right\rangle_F.\end{aligned}$$

We can now assemble this into the product of two terms—one depending on the realization only, and one depending on the sample autocovariances:

$$\begin{aligned}&(2\pi)^{md/2}\,\mathrm{P}\left((\mathbf{x}[t])_{t=p+1,\dots,m+p}\right)\\&=\exp\left(\sum_{t=p+1}^{m+p}\frac{1}{2}\left\|\mathbf{x}[t]\right\|^2\right)\\&\cdot\exp\left(m\sum_{s=1}^{p}\sum_{j=1}^{r}c_j\left\langle\mathbf{R}[s],\mathbf{D}_j[s]\right\rangle_F\right.\\&\left.-\frac{m}{2}\sum_{s,s'=1}^{p}\sum_{j,j'=1}^{r}c_jc_{j'}\left\langle\mathbf{D}_j[s],\mathbf{D}_{j'}[s']\mathbf{R}[s-s']\right\rangle_F\right)\end{aligned}$$

This concludes the proof. □

Proof of Proposition 5. We want to apply Theorem 1 of Bolte, Sabach, and Teboulle [6].

First, we require that (20) is a Kurdyka–Łojasiewicz (KL) function. For (20), H is a polynomial function and thus semi-algebraic; f is a norm of rational order and so semi-algebraic by [6, Example 4]; and g is the indicator function of a semi-algebraic set and so semi-algebraic. Then, we can invoke [6, Thm. 3], which says that any semi-algebraic function is also a KL function.

Next, we require that f and g are proper and lower semi-continuous and H is continuously differentiable. H is analytic and so continuously differentiable. f is continuous and so lower semi-continuous. As an indicator function, g is not continuous but is lower semi-continuous. Both f and g are proper functions.

Then, we must show the following: $\nabla_{\mathbf{D}_j}H$ and $\nabla_{\mathbf{c}_i}H$ are globally Lipschitz continuous, those Lipschitz constants are finite, and H is Lipschitz continuous on bounded subsets. The latter is trivially satisfied since $H\in C^2$ [6, Remark 3]. First, we derive the block Lipschitz constant of $\nabla_{\mathbf{D}_j}H$:

$$\begin{aligned}&\|\nabla_{\mathbf{D}_j}H\left(\mathbf{D}_1,\dots,\mathbf{D}_j,\dots,\mathbf{D}_r,\mathbf{c}_1,\dots,\mathbf{c}_n\right)\\&-\nabla_{\mathbf{D}_j}H\left(\mathbf{D}_1,\dots,\mathbf{D}'_j,\dots,\mathbf{D}_r,\mathbf{c}_1,\dots,\mathbf{c}_n\right)\|_F\\&\le\left\|\frac{1}{n}\sum_{i=1}^{n}c_{i,j}\mathbf{R}_i\right\|\cdot\left\|\mathbf{D}_j-\mathbf{D}'_j\right\|_F.\end{aligned}$$

Provided that $c_{i,j}<\infty$ and $\|\mathbf{R}_i\|<\infty$ for all $i=1,\dots,n$, the Lipschitz constant is finite. We can repeat the same argument for all $j=1,\dots,r$. Now, we derive the block Lipschitz constant of $\nabla_{\mathbf{c}_i}H$:

$$\begin{aligned}&\|\nabla_{\mathbf{c}_i}H\left(\mathbf{D}_1,\dots,\mathbf{D}_r,\mathbf{c}_1,\dots,\mathbf{c}_j,\dots,\mathbf{c}_n\right)\\&-\nabla_{\mathbf{c}_i}H\left(\mathbf{D}_1,\dots,\mathbf{D}_r,\mathbf{c}_1,\dots,\mathbf{c}'_j,\dots,\mathbf{c}_n\right)\|_F\\&\le\left\|\frac{1}{n}\mathbf{G}_i\right\|\cdot\left\|\mathbf{c}_i-\mathbf{c}'_i\right\|,\end{aligned}$$

where $[G_i]_{j,j'}=\left|\left\langle\mathbf{D}_j,\mathbf{R}_i\mathbf{D}_{j'}\right\rangle_F\right|$. The Lipschitz constant is finite if $\|\mathbf{R}_i\|<\infty$ since $\|\mathbf{G}_i\|\le\frac{1}{n}\|\mathbf{R}_i\|\cdot\sum_{j=1}^{r}\|\mathbf{D}_j\|_F^2=\frac{1}{n}\|\mathbf{R}_i\|<\infty$.

This concludes the proof as we can now apply [6, Thm. 1]. □