THE LINEAR PROCESS MIXTURE MODEL

Jason A. Palmer\(^1\), Ken Kreutz-Delgado\(^2\), and Scott Makeig\(^1\)

\(^1\)Swartz Center for Computational Neuroscience  
\(^2\)Department of Electrical and Computer Engineering  
University of California San Diego, La Jolla, CA 92093  
{japaler, kreutz, smakeig}@ucsd.edu

ABSTRACT
We consider a likelihood framework for analyzing multivariate time series as mixtures of independent linear processes. We propose a flexible, Newton algorithm for estimating impulse response functions associated with independent linear processes and an EM-based finite mixture model to handle intermittent regimes. Simulations and application to EEG are also provided.

1. INTRODUCTION
Multichannel blind deconvolution has been considered in some detail by Amari, Cichocki, Zhang, and others (see Cichocki and Amari (2002) [3, Chs. 9,10] and references), and in fact a Newton algorithm has been proposed [3, §10.3]. However the derivation of this algorithm has been somewhat tersely stated, and couched in what may seem daunting frameworks of Lie groups, non-holonomic spaces, and Riemannian manifolds. Likelihood has been used to derive estimating functions, but the likelihood is not commonly employed algorithmic developments involving time series. Here we consider a general likelihood framework for the analysis of mixtures of linear processes, similar to the framework considered by Amari [3, §10.3]. However we shall extend the framework in two main directions. First, we use the asymptotic segment likelihood derived to formulate a finite mixture model, and enable straightforward application to more complex Hidden Markov Models (HMMs). Second, we formulate the Newton algorithm with respect to the causal impulse response(s) associated with the mixed linear processes, rather than the coefficients of the unmixing system. Updating the “forward”, causal impulse response parameters directly facilitates a more general framework involving models with shared, or overlapping component processes. Thus we ultimately have in mind a “dictionary” of processes which may be “mixed-and-matched” to form complete models corresponding to different contexts. Inverse filters are constructed “on the fly” from a complete set of impulse responses, and possibly cached if recurring. Another significant product of this work is the formulation the Newton algorithm directly with respect to the likelihood, without introducing potentially unfamiliar transformations or filter decompositions. A steadfast application of principle leads through an apparent notational thicket to ultimately reveal the multivariate convolutive Newton algorithm in all its beautiful simplicity. Furthermore, the natural gradient is seen to arise as a natural simplification of the Newton algorithm.

We consider the analysis of observations, \(x(t) \in \mathbb{R}^n\), \(-\infty < t < \infty\), modeled as a (complete) linear superposition of independent source random processes, \(s_i(t), i = 1, \ldots, n\),

\[
x(t) = \sum_{i=1}^{n} s_i(t)
\]

In particular, we consider source processes that are causal vector random processes of the form,

\[
s_i(t) = \sum_{k=0}^{\infty} h_i(k) \varepsilon_i(t-k)
\]

where \(\sum_{k=0}^{\infty} ||h_i(k)|| < \infty\) (equivalently, \(\sum_k ||b_{ij}(k)|| < \infty\), \(j = 1, \ldots, n\)), and \(\varepsilon_i(t)\) is an i.i.d. process with zero mean and finite variance, \(E[\varepsilon_i(t)^2] < \infty\). We refer to \(h_i(k)\) as the impulse response associated with the process \(s_i(t)\).

We are primarily interested in linear processes whose infinite response results from a feedback process, suggesting an ARMA\((p_i, q_i)\) model of the form,

\[
s_i(t) = \sum_{j=1}^{p_i} A_i(j) s_i(t-j) + \sum_{k=0}^{q_i} b_i(k) \varepsilon_i(t-k)
\]

\(^1\)This definition of a causal linear process is consistent with Hannan (1970) [5, p. 209], who uses the term generalized linear process if the impulse response is square summable. Bartlett (1955) [1, p. 147] uses the term linear process for square summable impulse responses. Brockwell and Davis (1987) [2, p. 404], and Priestley (1981) [11, §3.5.7] use the weaker requirement that the innovations of a linear process be uncorrelated, with [2] requiring absolute summability, while [11] refers only to processes with square-summable impulse response, calling them “general linear processes”. Absolute summability of the impulse response, required here, ensures existence of the Fourier transform, and is equivalent to bounded-input bounded-output (BIBO) stability of the process [7, p. 30-31]. Independence of innovations implies complete stationarity [1].
In this case, the impulse response is identity generated from the following recursion, with \( h_i(\ell) = 0 \) for \( \ell < 0 \), \( h_i(0) = b_i(0) \), and
\[
\varepsilon(t) = \sum_{j=1}^{p} a_i(j)\varepsilon(t-j) + b_i(\ell), \quad \ell > 0
\]
(4)

In the case of a complete set of linear processes, \( s_i(t) \), \( i = 1, \ldots, n \), the respective impulse responses may be collected into a matrix series, \( H_k \triangleq [b_1(k) \cdot b_n(k)] \), and the innovations processes into a vector innovations process, \( \varepsilon(t) = [\varepsilon_1(t) \cdots \varepsilon_n(t)]^T \) with independent components,
\[
x(t) = \sum_{k=0}^{\infty} H_k \varepsilon(t-k)
\]
(5)

In general, the feedback systems and associated AR matrix coefficients, \( A_i(j), j = 1, \ldots, p_i \), differ for different linear processes. If the feedback systems are common, say \( A_{i_1} = \cdots = A_{i_k} \), then \( B_i(k) \triangleq [b_{i_1}(k) \cdots b_{i_k}(k)] \) and \( \varepsilon_i(t) \triangleq [\varepsilon_{i_1}(t) \cdots \varepsilon_{i_k}(t)] \) may be substituted for \( b_i(k) \) and \( \varepsilon_i(t) \) in (3). If the feedback coefficients are common to all the combined processes, then the usual vector ARMA process results,
\[
x(t) = \sum_{j=1}^{p} A(j) x(t-j) + \sum_{k=0}^{q} B(k) \varepsilon(t-k)
\]
(6)

The likelihood framework proposed here allows systematic testing of the fit of ARMA models with common and separate feedback processes of various orders using the generalized likelihood ratio test. Temporally independent innovations processes with dependent component subspaces can also be handled in the likelihood framework [8], but we focus here on innovations with independent components.

We use an instantaneous mixture model as this allows a reasonable compromise between model flexibility and complexity. More generally the segment likelihood may be used directly Markov dependence in model transitions, or continuous variation between model modes.

1.1. Notation

We use the following notation, where \( H \) and \( W \) are (generally two-sided) stable multivariate filters,
\[
(H * x)(t) \triangleq \sum_{k=-\infty}^{\infty} H_k x(t-k)
\]
\[
(W * H)_k \triangleq \sum_{\ell=-\infty}^{\infty} W_\ell H_{k-\ell}
\]

Note that \( W * (H * x) = (W * H) * x \triangleq W * H * x \). We also use the following notation,
\[
(H(-))(k) \triangleq H_{-k}, \quad (H^T)(k) \triangleq H_k^T
\]

Let \( \delta \) be the multivariate identity filter. If \( H \) is a causal stable multivariate filter, then it will generally have a two-sided stable inverse filter \( W = H^{-1} \), such that \( W * H = \delta \).

We note for reference the following Discrete Fourier Transform formulæ,
\[
H(\omega) \triangleq \sum_{k=-\infty}^{\infty} H_k e^{-i\omega k}, \quad H_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} H(\omega) e^{i\omega k} d\omega
\]

Also note the multivariate DFT convolution theorem,
\[
\frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega)Y(\omega) e^{i\omega k} d\omega = (X * Y)_k
\]
and the inverse theorem,
\[
W_\ell = \frac{1}{2\pi} \int_{-\pi}^{\pi} H(\omega)^{-1} e^{i\omega k} d\omega \Rightarrow W * H = \delta = H * W
\]

Note that multivariate convolution is not commutative in general.

2. THE LINEAR PROCESS MIXTURE MODEL

We shall take the data to consist of a set of \( N \) samples of segments of a certain length, \( T \). We regard the samples as independent, though in the case of a continuous record we shall consider all (overlapping) segments of length \( T \) to be samples, \( p(X_t; \Theta) = \prod_{t} p(X_t; \Theta) \).

Consider a finite mixture model of \( M \) segment models,
\[
p(X_t; \Theta) = \sum_{m=1}^{M} \gamma_m p(X_t|m; \Theta_m), \quad \sum_{m} \gamma_m = 1, \gamma_m > 0
\]

In our case \( \Theta_{\Theta_m} = H^{(m)} \), where \( H^{(m)} \) is the multivariate matrix impulse response associated with model \( m \). Mixture model parameters are updated according to the usual EM method [9] given the likelihood determined in the next section. The model superscript is suppressed in the sequel.

2.1. Asymptotic Segment Likelihood

Consider the likelihood of a segment of data, \( X_t \triangleq [x(t-T+1) \cdots x(t)] \), of length \( T \) samples, modeled as a multivariate causal linear process. Given a model, \( m \), using the asymptotic result for block Toeplitz matrices [10, 3, 4, 6] we have for the

\[\text{\footnotesize{\begin{enumerate}}\]
approximate likelihood,
\[ p(X_t | m; H) \approx \exp \left( -\frac{T}{2\pi} \int_0^{2\pi} \log |\det \sum_k H_k e^{-i\omega t}| d\omega \right) \]
\[
\times \prod_{\tau = 0}^{T-1} p_x(W + \Delta t(\tau - \tau)) \] (7)

Thus the approximate mean log likelihood is given by,
\[ L(H | X_t) = -\frac{1}{2\pi} \int_0^{2\pi} \log |\det \sum_{k=0}^{\infty} H_k e^{-i\omega t}| d\omega \]
\[
+ \frac{1}{T} \sum_{\tau = 0}^{T-1} \log p_x(W + \Delta t(\tau - \tau)) \] (8)

The integral is readily computed approximately using the FFT to produce the sampled matrix frequency response, and summing their log determinants as a Riemann approximation.

3. THE LIKELIHOOD GRADIENT

3.1. Real Derivatives of Likelihood Values

The Real derivative of a complex valued function of a Real variable, \( f : \mathbb{R} \rightarrow \mathbb{C} \) is defined by,
\[ \frac{df}{dx}(x) \triangleq \frac{df}{dx}(\Re(x)) + i \frac{df}{dx}(\Im(x)) \] (9)

with Real derivatives of complex matrix valued functions defined similarly. The Real derivative of a product of complex matrices \( U, V : \mathbb{R} \rightarrow \mathbb{C}^{n \times n} \) satisfies the following product rule:
\[ \frac{d}{dx}(UV) = \left( \frac{d}{dx}U \right)V + U \left( \frac{d}{dx}V \right) \]

Using the product rule we have,
\[ \frac{d}{dx}(CC^{-1}) = \left( \frac{d}{dx}C \right)C^{-1} + C \left( \frac{d}{dx}C^{-1} \right) = 0 \]

which yields the following formula for the Real derivative of the matrix inverse,
\[ \frac{d}{dx}C^{-1} = -C^{-1} \left( \frac{d}{dx}C \right)C^{-1} \] (10)

Using this formula, we have in particular,
\[ \frac{\partial}{\partial H_{kij}} W(\omega) = \frac{\partial}{\partial H_{kij}} \left( \sum t \ H_t e^{-i\omega t} \right)^{-1} = \]
\[ - \left( \sum t \ H_t e^{-i\omega t} \right)^{-1} E_{ij} \left( \sum t \ H_t e^{-i\omega t} \right)^{-1} e^{-i\omega k} \] (11)

where \( E_{ij} \) is the matrix with 1 in the (i, j)th element and 0 elsewhere. We thus have the following (real) formula,
\[ \frac{\partial}{\partial H_{kij}} W_t = -\frac{1}{2\pi} \int_0^{2\pi} W(\omega) E_{ij} W(\omega) e^{i\omega \tau} e^{-i\omega k} d\omega \]
\[ = -\sum_{\tau = -\infty}^\infty W_t E_{ij} W_{t-k-\tau} \] (12)

3.2. Complex Derivatives and the Chain Rule

If \( g : \mathbb{C}^{n \times n} \rightarrow \mathbb{R} \) is a real valued function of a complex matrix, and \( C : \mathbb{R}^{n \times n} \rightarrow \mathbb{C}^{n \times n} \) is a complex matrix valued function of a real matrix, then we have the following specialized chain rule:\footnote{The complex Wirtinger derivatives are defined by,
\[ \frac{\partial g(z)}{\partial z} \triangleq \frac{1}{2} \left( \frac{\partial g(z)}{\partial \Re z} - i \frac{\partial g(z)}{\partial \Im z} \right), \quad \frac{\partial g(z)}{\partial \bar{z}} \triangleq \frac{1}{2} \left( \frac{\partial g(z)}{\partial \Re z} + i \frac{\partial g(z)}{\partial \Im z} \right) \]}
\[ \frac{\partial g(C)}{\partial B_{ij}} = \text{tr} \left( \frac{\partial g}{\partial C} \frac{\partial C^T}{\partial B_{ij}} + \frac{\partial g}{\partial C} \frac{\partial C^T}{\partial B_{ij}} \right) \]
\[ = 2 \text{ Re} \text{ tr} \left( \frac{\partial g}{\partial C} \frac{\partial C^T}{\partial B_{ij}} \right) \] (13)

since \( \frac{\partial g}{\partial C} = \left( \frac{\partial g}{\partial C} \right)^* \) and \( \frac{\partial C^*}{\partial B_{ij}} = \left( \frac{\partial C}{\partial B_{ij}} \right)^* \) according to our assumptions. Note that the first derivative in (13) is a complex derivative and the second is a real derivative.

Now, using the fact that for the complex derivative,
\[ \frac{\partial}{\partial C} \log \det CC^H = C^{-T} \]

along with the chain rule (13) and the formula (11), we have the following,
\[ \frac{\partial}{\partial H_{kij}} \log |\det \sum t H_t e^{-i\omega t}| \]
\[ = \frac{\partial}{\partial H_{kij}} \frac{1}{2\pi} \int_0^{2\pi} \log |\det \sum t H_t e^{-i\omega t}| d\omega \]
\[ = \text{Re} \frac{1}{2\pi} \int_0^{2\pi} \left( \sum t H_t e^{-i\omega t} \right)^{-T} E_{ij} e^{-i\omega k} d\omega \]

3.3. Likelihood Gradient

Using the formulae derived in the previous sections, we can evaluate the derivatives of the log determinant integral terms in the likelihood \( L \) in (8),
\[ \frac{\partial}{\partial H} \frac{1}{2\pi} \int_0^{2\pi} \log |\det \sum t H_t e^{-i\omega t}| d\omega \]
\[ = \text{Re} \frac{1}{2\pi} \int_0^{2\pi} \left( \sum t H_t e^{-i\omega t} \right)^{-T} e^{-i\omega k} d\omega = W_{-k} \]

where \( W_k \) is the kth matrix in the inverse filter \( W = H^{-1} \).

Define the function,
\[ \varphi(\varepsilon(t)) \triangleq \log p_x(\varepsilon(t)) = \log p_x(\sum_k W_k \varepsilon(t-k)) \]
Now,
\[
\frac{\partial}{\partial H_{kij}} \varphi(e(t)) = \sum_{\ell = -\infty}^{\infty} \text{tr} \left( \frac{\partial \varphi}{\partial W_{\ell}} \frac{\partial W_{\ell}^T}{\partial H_{kij}} \right)
\]
\[
= -\sum_{\ell = -\infty}^{\infty} \text{tr} \nabla \varphi(e(t)) x(t - \ell) \nabla \varphi(e(t)) (\sum_{\tau} \tau_{\tau} W_{\tau} e_{\tau - k - \tau}^T)
\]
So that,
\[
\frac{\partial}{\partial H_k} \varphi(e(t)) = -\sum_{\tau = -\infty}^{\infty} W_{\tau}^T \nabla \varphi(e(t)) e(t - k - \tau)^T
\]
Putting all this together, we get the gradient formula,
\[
\frac{\partial L}{\partial H_k} = -W_{-k}^T - \sum_{\ell = -\infty}^{\infty} W_{\tau}^T E \{ \nabla \varphi(e(t)) e(t - k - \ell)^T \}
\]
Define,
\[
\Phi = E \{ \nabla \varphi(e(t + \tau)) e(t)^T \}
\]
Then we have,
\[
\frac{\partial L}{\partial H_k} = -W_{-k}^T - (W_{-k}^T \Phi)_k, \quad k = 0, \ldots, q
\]
This formula can be expressed simply as the following real filter gradient,
\[
\frac{\partial L}{\partial H} = -W_{-k}^T - W_{-k}^T \Phi = -W_{-k}^T (\delta + \Phi)
\]

4. HESSIAN AND NEWTON METHOD

The Newton method for optimization essentially makes a quadratic approximation of the cost function, iteratively moving in the direction of the computable optimum of this approximation (a stationary point, optimal assuming definiteness of the Hessian). Specifically, the Newton method makes the approximation,
\[
L(A + dA) \approx L(A) + \left( \frac{\partial L}{\partial A}, dA \right) + \frac{1}{2} \left( dA, \nabla^2 L(dA) \right)
\]
where the gradient and Hessian are evaluated at a current iterate, \(A^{(t)}\). The optimal direction is found by solving the Newton equation,
\[
-\frac{\partial L}{\partial A} = \nabla^2 L(dA)
\]
for \(dA\), and putting \(A^{(t+1)} = A^{(t)} + \alpha dA\), with \(0 < \alpha \leq 1\).

In the multivariate real filter case, we have,
\[
\langle A, B \rangle = \sum_{\tau = -\infty}^{\infty} \text{tr}(A_{\tau} B_{\tau}^T)
\]
To compute the second differentials, we will require the formula,
\[
\frac{\partial}{\partial H_{kij}} e(t) = \sum_{\ell = -\infty}^{\infty} \left( \frac{\partial}{\partial H_{kij}} W_{\ell} \right) x(t - \ell)
\]
\[
= -\sum_{\ell = -\infty}^{\infty} \sum_{\tau = -\infty}^{\infty} W_{\tau} e_{\tau - k - \tau} x(t - \ell)
\]
Now, we first compute using the product rule for ordinary differentiation,
\[
\frac{\partial}{\partial H_{uij}} \frac{\partial L}{\partial H_k} = -\left( -\sum_{\tau} W_{\tau} e_{\tau - u - \tau} \right)^T
\]
\[
+ \sum_{\tau} \sum_{\ell} \sum_{\ell} W_{\tau}^T \text{diag}(\varphi''(e_{\ell})) W_{\ell} e_{\tau - u - \ell} e_{\ell - k - \tau}^T
\]
\[
+ \sum_{\tau} \sum_{\ell} W_{\tau}^T \nabla \varphi(e_{\ell}) \left( W_{\ell} e_{\tau - u - \ell} e_{\ell - k - \tau}^T \right)
\]
The second differential is then determined as follows,
\[
\langle H(dH) \rangle
\]
\[
= \sum_{u = 0}^{\infty} \sum_{\tau} W_{-u - \tau - k}^T dH_u W_{-u - \tau - k}^T
\]
\[
+ \sum_{u = 0}^{\infty} \sum_{\tau} \sum_{\ell} W_{-u - \ell}^T dH_u W_{\ell}^T \nabla \varphi(e_{\ell}) e_{\ell - k - \tau}^T
\]
\[
+ \sum_{u = 0}^{\infty} \sum_{\tau} \sum_{\ell} W_{-u - \ell}^T \text{diag}(\varphi''(e_{\ell})) W_{\ell} dH_u e_{-u - \ell} e_{\ell - k - \tau}^T
\]
\[
+ \sum_{u = 0}^{\infty} \sum_{\tau} \sum_{\ell} W_{-u - \ell}^T \nabla \varphi(e_{\ell}) e_{-u - \ell - k - \tau}^T dH_u W_{\ell}^T
\]
\[
= \left( W^T * dH^T * W \right)_{-k} + \left( W^T * dH^T * W \right)_{-k}
\]
\[
+ \left( W^T * W \right)_{-k} + \sum_{u = 0}^{\infty} \sum_{\tau} \sum_{\ell} W_{\ell - \tau - u - \tau}^T \text{diag}(\varphi''(e_{\ell})) W_{\ell - \tau - u - \tau} e_{\ell - k - \tau}^T
\]
The last summation can be rewritten,
\[
\sum_{\tau = -\infty}^{\infty} \sum_{\ell = -\infty}^{\infty} W_{\tau - \ell - u - \tau}^T \text{diag}(\varphi''(e_{\ell})) W_{\ell - \tau - u} e_{\ell - \tau - u - \tau}^T
\]
Taking expectations and using the fact that the innovations are
Thus we have, substituting
\[
\sum_{\tau=-\infty}^{\infty} W_{\tau-k}^T E\{\text{diag}(\varphi''(\varepsilon_i)) (W * d\hat{H})_\tau \varepsilon_{t-\tau} \varepsilon_{t-\tau}^T\}
\]
\[= \sum_{\tau=-\infty}^{\infty} W_{\tau-k}^T (F(W * dH))_\tau\]
\[= (W_{\tau-k}^T F(W * dH))_k\]
where the matrix filter function \(F\) consists of Hadamard products,
\[
(F(B))_{kj} \leq \left\{ \begin{array}{ll} E\{\varphi''(\varepsilon_i)\varepsilon_i^2\} B_{0ij}, & k = 0, i = j \\ E\{\varphi''(\varepsilon_i)\} E\{\varepsilon_i^2\} B_{kij}, & \text{otherwise} \end{array} \right.
\]
(18)
Thus we have, substituting \(\Phi_{\tau} \triangleq E\{\nabla \varphi(\varepsilon_{t+\tau})\varepsilon_t^T\},\)
\[
(\mathcal{H}(dH))_k = \left( W^T * d\hat{H}^T * W^T \right)_{-k} + (H^T * dW^T * W^T + \Phi_{-})_{-k}
\]
\[+ (W^T * \Phi_{-} * d\hat{H}^T * W^T)_{-k} + (W^T * F(W * dH))_k
\]
Now let us make the following definition,
\[
d\hat{H} \triangleq W * dH
\]
Then the second differential can be written,
\[
\mathcal{H}(dH) = W^T * d\hat{H}^T + W^T * d\hat{H}^T * \Phi
\]
\[+ W^T * \Phi * d\hat{H}^T + W^T * F(d\hat{H})
\]
Recall that the matrix filter gradient is given by,
\[
G = \frac{\partial L}{\partial \hat{H}} = -W^T * (\delta + \Phi)
\]
Thus the Newton equation can be written,
\[
\delta + \Phi = d\hat{H}^T + d\hat{H}^T * \Phi + \Phi * d\hat{H}^T + F(d\hat{H})
\]
This equation can be further simplified by approximating \(\Phi\) in the Hessian transformation by its value at the solution, where the innovations \(\varepsilon(t)\) are independent, which leads to,
\[
\delta + \Phi = F(d\hat{H}) + G(d\hat{H}^T)
\]
where we define \(G\) as the constant matrix filter Hadamard products,
\[
(G(A))_{kij} \triangleq (\lambda_i + \lambda_j + 1) A_{kij}
\]
using the definition,
\[
\lambda_i \triangleq E\{\varphi'''(\varepsilon_i)\varepsilon_i\}
\]
(19)
Let us also define,
\[
\eta_i \triangleq E\{\varphi''(\varepsilon_i)\varepsilon_i^2\}, \quad \kappa_i \triangleq E\{\varphi''(\varepsilon_i)\}, \quad \sigma_i^2 \triangleq E\{\varepsilon_i^2\}
\]
(20)
We can then solve for \(d\hat{H}\) using the reduced equations,
\[
\lambda_i + 1 = (\eta_i + 2\lambda_i + 1) d\hat{H}_{0ii}, \quad i = 1, \ldots, n
\]
(21)
and for \(k = 0, i \neq j,\) and \(k \neq 0,\)
\[
\begin{bmatrix}
\Phi_{kij} \\
\Phi_{-kj}
\end{bmatrix}
\triangleq
\begin{bmatrix}
\kappa_i \sigma_i^2 \\
\kappa_j \sigma_j^2
\end{bmatrix}
\begin{bmatrix}
(\lambda_i + \lambda_j + 1) \\
(\lambda_i + \lambda_j + 1)
\end{bmatrix}
\begin{bmatrix}
d\hat{H}_{kij} \\
d\hat{H}_{-kji}
\end{bmatrix}
\]
(22)
The filter update \(dH\) is then determined by,
\[
dH = \hat{H} * d\hat{H}
\]
(23)
The stability conditions are also seen to be equivalent to the instantaneous case [3].

### 5. EXPERIMENTS

As an example simulation, we considered a two-channel time series consisting of two alternating convolutive mixing systems given by two distinct ARMA(4,4) processes with i.i.d. Laplacian innovations. The two impulse responses associated with the two systems are shown in Figures 1(a) and 1(b). We generated 30,000 samples from each system, dividing these into two 15,000 sample segments, and constructed the observed series as an alternating concatenation of these four continuous segments. We treated the data as a set of approximately 59,000 segments of length 1000, and fixed the learned impulse response length to 40. We initialized the impulse responses with approximate pure delay (identity) filters of delay 5 samples. We began with 20 natural gradient steps with the idea of first getting near the optimal basin, and then switched to the Newton updates, which required only limited additional computation. The Newton method converged after 50 iterations, while continuing the natural gradient was found to require at least four times as many (costly) iterations to achieve similar performance.

We also applied the algorithm to a real time series derived from an instantaneous ICA unmixing of EEG data. In this example we analyzed a single alpha (10 Hz) component, performing automatic segmentation of the series using two models. Figure 2(a) shows the power spectrum of the entire series, and Figures 2(b) and 2(c) show the spectra of data segmented by maximum a posteriori likelihood into alpha and non-alpha periods respectively.

### 6. DISCUSSION

We found a distinct advantage in convergence robustness and speed using the approximate natural gradient. Few iterations are required for convergence from random initialization, however iteration time, particularly in evaluating/updating non-linear covariance estimates for each segment, remains a significant limiting factor. The method can be parallelized over models and segments, with
Fig. 1. Simulation experiment. Two causal ARMA(4,4) mixing systems with impulses shown in (a) and (b) were mixed with i.i.d. Laplacian innovations. The learned impulse responses, shown in (c) and (d), are scaled delayed versions of the mixing systems. (e) The data is accurately segmented by posterior likelihood. (f) The Newton method converges much more rapidly than the natural gradient.

the latter being key to reducing iteration time. We also found that while convergence is robust, it does appear to depend on initialization with a delayed approximate identity filter can be crucial to prevent convergence problems. If a zero-lag identity is used as the filter initializations, the algorithm tends to converge to the minimum phase “tail” of the impulse response, being unable to “shift” the current estimate when assuming a fixed origin at the start of the filter.

7. REFERENCES


