Efficient Class-Specific Models for Autoregressive Processes with Slowly Varying Amplitude in White Noise

Dr. Paul M. Baggenstoss *†
Naval Undersea Warfare Center
Newport RI, 02841
401-832-8240 (TEL)
p.m.baggenstoss@ieee.org
http://www.npt.nuwc.navy.mil/csf

July 13, 2007

Abstract

This paper describes an efficient model to describe an autoregressive signal with slowly-varying amplitude in additive white Gaussian noise. Even a simple low-order autoregressive model becomes complicated by varying amplitude and additive white noise. However, by approximating the signal amplitude as piecewise-constant, an efficient filtering approach can be applied in order to compute the maximum likelihood estimate for the entire data record. The model is efficient both in terms of havng a compact set of parameters and in the computational sense. Simulation results are provided. The algorithm has applications in signal modeling for underwater acoustic signals, particularly active wideband signals such as explosive sources.

1 Introduction

In narrow-band active sonar, the processing bandwith is too narrow to permit the observation of variations in the signal plus interference (SI) spectrum. This is not true in wide-band systems where the power spectrum of interference, especially ambient noise, may significantly differ from the signal and reverberation. After the data has been filtered to pre-whiten the interference, signal can be modeled as a colored Gaussian process with fixed spectral shape, but fluctuating amplitude, in additive white Gaussian noise (WGN). Unless the signal and interference have the

*This work was supported by the Office of Naval Research (contr nr. N0001405WR20125).
†Unclassified. Approved for public release. Distribution unlimited.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
13 JUL 2007		**00-00-2007 to 00-00-2007**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Efficient Class-Specific Models for Autoregressive Processes with Slowly Varying Amplitude in White Noise	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Undersea Warfare Center,Newport,RI,02841**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**34**	

same spectral shape, the fluctuating signal amplitude causes the SI spectrum to vary. This in turn requires a high-fidelity model to estimate the continually changing SI spectrum, causing over-parameterization. In this paper we describe a compact model that parameterizes the signal spectum as a constant autoregressive (AR) process with fluctuating amplitude. The signal AR parameters, white noise variance, and parameters of the amplitude envelope function constitute a complete set of model parameters. We present a very efficient means of computing the maximum likelihood (ML) estimates for these parameters based on filtering. We also report on the performance of the models using simulated data and show how the model can be used in a class-specific classifier.

1.1 Previous Work

The problem that occurs when white noise is added to an AR process, thereby forming an effective ARMA process, is well known [1] and numerous methods exist for determining the underlying AR parameters [2], [3]. Previous work did not consider varying AR process amplitude. As the AR process varies in level, it naturally changes the ARMA parameters which further complicates the problem. But at the same time, it opens up an opportunity for great simplification since the underlying parameters are compact, involving only the AR parameters, the white noise variance, plus the parameters of the changing AR process amplitude.

1.2 Paper Summary

1. In section 2, we present the mathematical model for an autoregressive process modulated by an envelope function in additive white Gaussian noise.

2. In section 3, we talk about how the model may be used in a classifier.

3. In section 4, we talk in general about how to estimate the model parameters.

4. In section 5, we present the exact likelihood function, which is not practical to use but serves as a standard.

5. In section 6, we present a simplifying assumption that the envelope function varies slowly so that if the data is segmented, the data in a segment has a fixed spectral model. Then, we analyze in detail the PDF of a segment of data. We find the segment power spectrum and autocorrelation function (section 6.1), derive the exact PDF of a segment (section 6.2), develop an equivalent autoregressive moving average (ARMA) model for the segment (section 6.3), find the exact derivatives and Fisher information matrix (FIM) of the PDF (sections 6.4, 6.5). We then present a frequency-domain (FD) approximation to the segment PDF (section 6.6), obtain the derivatives and FIM (sections 6.7, 6.8). We use the FD approximation to derive a filtering approach to computing the segment PDF (sections 6.9, 6.10, 6.11), as well as the derivatives (section 6.12).

6. In section 7, we show how the segment PDF results can be combined to obtain the PDF and derivatives for the entire data record.

7. In section 8, we summarize the algorithhm.

8. In section 9, we provide simulation results.

2 Problem Formulation

In this section we formulate mathematically an autoregressive (AR) process with slowly-varying power in WGN.

2.1 Data model

Consider an AR signal process y_t, of order P:

$$y_t = -\sum_{i=1}^{P} a_i \, y_{t-i} + e_t, \tag{1}$$

where e_t is a zero-mean independent Gaussian innovation process with fixed variance σ^2. Let the data be modulated by a time-varying signal power function λ_t and corrupted by u_t, a zero-mean

independent Gaussian additive noise process with variance σ_n^2:

$$x_t = \lambda_t^{1/2}(\boldsymbol{\phi})\, y_t + u_t,\ t = 1, 2 \ldots N, \tag{2}$$

where $\boldsymbol{\phi}$ are the parameters of the envelope function (assumed to be of dimension Q). Note that both σ^2 and $\lambda_t(\boldsymbol{\phi})$ affect the power of the process at time t. In order to prevent redundant parameters, we assume that $\boldsymbol{\phi}$ is normalized as follows:

$$\sum_{t=1}^{N} \lambda_t(\boldsymbol{\phi}) = 1. \tag{3}$$

The complete $(P + Q + 2)$-dimensional set of model parameters are

$$\boldsymbol{\Theta} = [\sigma_n^2, \sigma^2, a_1, a_2 \ldots a_P, \boldsymbol{\phi}].$$

3 Classifier Methodology

Let $\mathbf{x} = [x_1, x_2 \ldots x_N]'$ be an observation vector of N samples from x_t. The goal of the classifier is to find the MAP classification hypothesis for explaining $\mathbf{x}$ as arising from one of several class hypotheses, H_i.

$$H_{\text{MAP}} = \text{argmax}\ p(H_i|\mathbf{x}) = \text{argmax}\ p(\mathbf{x}|H_i)\ p(H_i)$$

(Usually, we also assume that the apriori probablility of each classification is equal, so that the hypothesis is also a ML hypothesis. $H_{ML} = \text{argmax}\ p(\mathbf{x}|H_i)$.)

3.1 The Class Specific Classifier

The class-specific classifier is fundamentally a Bayesian classifier that produces a maximum a posteriori classification hypothesis given the observation of the raw data.

Since $\mathbf{x}$ is of very high dimension, a closed-form description of $p(\mathbf{x}|H_i)$ is impractical. The classical classification approach is to choose a set of features $\mathbf{z} = T(\mathbf{x})$ such that $\mathbf{z}$ is an approximately sufficient statistic for distinguishing between any pair of classification hypotheses, empirically estimate $p(\mathbf{z}|H_i)$ from training data, and output the MAP hypothesis given $\mathbf{z}$. As the number of

classification hypotheses, M, increases, the dimension of $\mathbf{z}$ must also increase to maintain sufficiency. But as the dimension of $\mathbf{z}$ increases, the accuracy of the estimated $p(\mathbf{z}|H_i)$ decreases. This tradeoff is the curse of dimensionality inherent to this classification method.

A class specific classifier solves the problem of the high dimensional $\mathbf{x}$ in a way that avoids the curse of dimensionality w.r.t. increasing M - by projecting the estimated $p(\mathbf{z}|H_i)$ into the raw data space, using a class-dependent reference hypothesis, $H_{0,i}$:

$$p(\mathbf{x}|H_i) = J_i(\mathbf{x})\ p(\mathbf{z}|H_i), \tag{4}$$

where

$$J_i(\mathbf{x}) = \frac{p(\mathbf{x}|H_{0,i})}{p(\mathbf{z}|H_{0,i})} \tag{5}$$

is called the "J-function". A different $\mathbf{z}_i = T_i(\mathbf{x})$ may be tailored to each classification hypothesis, H_i, and $\mathbf{z}_i$ only need be an approximately sufficient statistic for distinguishing between $H_{0,i}$ and H_i. Thus, the dimension of $\mathbf{z}_i$ does not depend on the number of classification hypotheses.

In this paper, we concentrate on a specific model with a particular set of features. The model is intended to model signals with a very specific form. By deriving the class-specific J-function, we are able to use this model in a classifier encompasing other models and features.

3.2 Approach to J-function

A number of strategies exist for implementation of $J_i(\mathbf{x})$ in (4). Either a fixed or floating reference hypotheses may be used for $H_{0,i}$ [4]. The ML method is a subset of the floating reference hypothesis method [4] and is prefereble whenever a model depends on a compact set of parameters and is suitable for ML exstimation. The maximum likelihood method, which we use here to implement $J_i(\mathbf{x})$, is written

$$J_i(\mathbf{x}) = \frac{p(\mathbf{x};\hat{\mathbf{\Theta}})}{(2\pi)^{-\frac{D}{2}}\ |\mathbf{I}(\hat{\mathbf{\Theta}})|^{\frac{1}{2}}}, \tag{6}$$

where $\hat{\mathbf{\Theta}}$ is the ML estimate of $\mathbf{\Theta}$, $\mathbf{I}(\hat{\mathbf{\Theta}})$ is the Fisher's information matrix [1] for the ML estimator of parameter $\mathbf{\Theta}$, and D is the dimension of $\mathbf{\Theta}$. Notice that to implement (6), we will need a functional form of the likelihood function as well as the Fisher's information matrix.

4 Parameter Estimation

Besides obtaining an efficient functional form of the likelihood function, we also need to estimate the values of the parameters. Our approach is to obtain initial parameter estimates for $\mathbf{\Theta}$, then iterate to find the ML estimate using an approximate Newton-Raphson iteration based on the Fisher's Information matrix.

The Newton-Raphson iteration requires not only the FIM, but also the first derivatives of the log-likelihood function with respect to the parameters. The first derivatives are more important from an accuracy point of view since an inaccurate FIM only slows down the algorithm, while an inacurate derivative gets you the wrong result.

Consider a general log PDF that depends on D parameters: $\boldsymbol{\theta} = [\theta_1, \theta_2 \ldots \theta_D]'$. The Fisher's information between any two parameters θ_i and θ_j is defined by

$$I_{\theta_i,\theta_j} = -\mathrm{E}\left\{\frac{\partial^2 \log p(\mathbf{x};\boldsymbol{\theta})}{\partial\theta_i \partial\theta_j}\right\}. \tag{7}$$

Collecting all these values into the matrix $\mathbf{I}(\boldsymbol{\theta})$, we have the $D \times D$ Fisher's information matrix. The Cramer-Rao lower bound states that the covariance matrix $\mathbf{C}$ of any joint unbiased estimator for the parameters $\boldsymbol{\theta}$ is such that

$$\mathbf{x}'\left(\mathbf{C} - \mathbf{I}^{-1}(\boldsymbol{\theta})\right)\mathbf{x} > 0$$

for all $\mathbf{x} \neq 0$. This effectively means that $\mathbf{I}^{-1}(\boldsymbol{\theta})$ is the lower bound for the covariance of any unbiased estimator.

The inverse of the Fisher's information matrix is a good estimate of the parameter estimation error covariance and is useful for iterative optimization. Given a parameter estimate $\boldsymbol{\theta}_n$, the new estimate is obtained as

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n + \mathbf{I}^{-1}(\boldsymbol{\theta}_n)\,\boldsymbol{\delta}, \tag{8}$$

where

$$\boldsymbol{\delta} = [d(\theta_1)\; d(\theta_2) \ldots]'$$

is the gradient vector formed from the first partial derivatives

$$d(\theta_i) \triangleq \frac{\partial}{\partial \theta_i} \log p(\mathbf{x}; \boldsymbol{\theta}) \Big|_{\theta_i = \theta_{n,i}}.$$

5 The Exact Likelihood Function.

To implement the numerator of (6), we need $p(\mathbf{x}; \boldsymbol{\Theta})$, the formula for the data PDF as a function of the parameters, or *likelihood function*. Let $\sigma^2\ \mathbf{R}^N$ be the covariance matrix of process y_t. Thus, $\mathbf{R}^N$ is the covariance matrix of the AR process y_t when $\sigma^2 = 1$. The covariance of a stationary AR process is symmetric and Toeplitz and can easily be derived from the inverse Fourier transform of the theoretical AR power spectrum [1]. Let the matrix $\boldsymbol{\Lambda}$ be a diagonal matrix with elements $\boldsymbol{\Lambda}_{t,t} = \lambda_t,\ 1 \leq t \leq N$. Let $\mathbf{u}$ be a zero-mean Gaussian random vector with variance σ_n^2. In matrix form, (2) becomes

$$\mathbf{x} = \boldsymbol{\Lambda}^{1/2}\mathbf{y} + \mathbf{u},$$

and the covariance of $\mathbf{x}$ is

$$\mathbf{C} \triangleq \mathrm{E}\{\mathbf{x}\mathbf{x}'\} = \sigma_n^2\mathbf{I} + \sigma^2\ \boldsymbol{\Lambda}^{1/2}\ \mathbf{R}^N\ \boldsymbol{\Lambda}^{1/2}. \tag{9}$$

Now we have the closed form for the PDF of x parameterized by $\mathbf{C}$, $p(\mathbf{x}; \mathbf{C})$.

$$\log p(\mathbf{x}; \boldsymbol{\Theta}) = -\frac{N}{2}\log(2\pi) - \frac{1}{2}\log|\det \mathbf{C}| - \frac{1}{2}\mathbf{x}'\mathbf{C}^{-1}\ \mathbf{x}. \tag{10}$$

Equation (10) is exact, but its usefulness is limited because the size of $\mathbf{C}$ is $N \times N$. In many real-world problems, N is too large to evaluate (10) efficiently. Nevertheless, (10) does serve an important role in evaluating the accuracy of approximate methods.

6 Piecewise Stationary Model

To arrive at an efficient implementation of the PDF, we assume that λ_t varies slowly with time, and so may be approximated by a constant, $L_m(\boldsymbol{\phi})$, over an interval of M samples, where $1 < M << N$.

Thus,

$$\lambda_t = L_m, \quad 1 + (m-1)M \leq t \leq mM, \tag{11}$$

where m is the segment number and we have dropped the argument ($\boldsymbol{\phi}$) for notation simplicity. This assumption means there is a fixed spectral model in effect in the segment. This leads to a simplified model for the PDF in a segment. Note that we **do not** assume the segments are independent when we compute the PDF of the entire data record. We only recognize that the model is constant within a segment.

We focus now on the PDF for a single data segment $\mathbf{x}_m$ and will later extend the result to the full data record $\mathbf{x}$. Let

$$\mathbf{x}_m = [x_{M(m-1)+1} \ldots x_{Mm}]'$$

be the segment m data. In this section, we concentrate now on the PDF of the segment m data, for which the parameters are defined as

$$\boldsymbol{\theta}_m = [\sigma_m^2, \sigma_n^2, a_1, \ldots a_P] \tag{12}$$

where we combine the parameters L_m and σ^2 into a single parameter

$$\sigma_m^2 = L_m \sigma^2. \tag{13}$$

6.1 Segment Power Spectrum and Autocorrelation Function

The power spectrum of the data within a segment can be written as

$$\rho_{k,m}^N = \sigma_n^2 + \frac{\sigma_m^2}{|A_k^N|^2}. \tag{14}$$

where we have used (2), (11), (13), and where A_k^N is shorthand for $A(e^{j2\pi k/N})$, which is the Z-transform of the AR polynomial $\mathbf{a} = [1, a_1, \ldots a_P]$ evaluated on the unit circle.

Since it is zero-mean, its covariance matrix is equal to its autocorrelation matrix. The autocorrelation function (ACF) is the inverse Fourier transform of the power spectrum. Since it is not practical to use the continuous frequency values we must find a practical means of computing the

ACF using the DFT. Let $r_{t,m}$ be the ACF lag t in segment m. For any stable stationary Gaussian process, the ACF decays to zero. Assuming $r_{t,m}$ dies to zero for $t < K$ where $K < M$, we can calculate $r_{t,m}$ from $\rho_{k,m}^{2M}$ using a length $2M$ inverse DFT. We first take the length $2M$ DFT of $\mathbf{a}$ (zero padded to length $2M$), compute $\rho_{k,m}^{2M}$ using (14), then take the inverse DFT. The result is valid up to lag M. In MATLAB,

```
A=fft([a(:); zeros(2*M-P-1,1)]);
rho = sig2n + sig2m./abs(A).^2;
rm = real(ifft(rho));
rm=rm(1:M);
```

6.2 Segment PDF

Modifying (10) for a single segment m,

$$\log p(\mathbf{x}_m; \mathbf{\Theta}) = -\frac{M}{2}\log(2\pi) - \frac{1}{2}\log|\det \mathbf{C}_m| - \frac{1}{2}\mathbf{x}_m' \mathbf{C}_m^{-1}\, \mathbf{x}_m, \tag{15}$$

where $\mathbf{C}_m$ is the $M \times M$ covariance

$$\mathbf{C}_m = \sigma_n^2 \mathbf{I} + \sigma_m^2 \mathbf{R}^M.$$

Since $\mathbf{C}_m$ is symmetric and Toeplitz, we may use the efficient Levinson algorithm to compute (15), which provides the determinant of $\mathbf{C}_m$ as a by-product, however M may still be too large to be practical.

6.3 ARMA model using Spectral Factorization

Although (14) is a compact spectral model, it is not in the most useful form. If we re-write it in a rational form, we may implement the PDF using linear filtering. By assuming the process is quasi-stationary within segments of length M, we may represent the power spectrum in a rational form equivalent to an ARMA process. We may re-write (14) as

$$\rho_{k,m}^N = \frac{\sigma_m^2 + \sigma_n^2 |A_k^N|^2}{|A_k^N|^2} = \sigma_{b,m}^2 \frac{|B_{k,m}^N|^2}{|A_k^N|^2}, \tag{16}$$

which is the PSD of an ARMA(P, P) process. By equating the numerators in (16) we see that the z-transformed AR and MA parameters, $A(z)$ and $B_m(z)$, respectively, are related by

$$\sigma_{b,m}^2 \; B_m(z)B_m^*(1/z^*) = \sigma_m^2 + \sigma_n^2 A(z)A^*(1/z^*). \tag{17}$$

To obtain the equivalent ARMA parameterization, we need to find $\sigma_{b,m}^2$ and $\mathbf{b}_m = [1, b_{m,1} \ldots b_{m,P}]$, the filter coefficients corresponding to $B_m(z)$. Using (17), we can solve easily for the coefficients of the order $2P+1$ polynomial in z corresponding to $\sigma_{b,m}^2 \; B_m(z)B_m^*(1/z^*)$, which we denote by $\boldsymbol{\beta}_m$. In MATLAB, equation (17) is realized in the time domain as:

```
beta_m = sig2n*conv(a(:),flipud(a(:)));
beta_m(P+1) = beta_m(P+1) + sig2m;
```

Using the technique of spectral factorization [5], we observe that $\boldsymbol{\beta}_m$ is proportional to the convolution of $\mathbf{b}_m$ and $\mathbf{b}_m^*$, which is $\mathbf{b}_m$ reversed in time so it has the combined roots of $\mathbf{b}_m$ and $\mathbf{b}_m^*$. The roots of $\mathbf{b}_m^*$ are the reciprocal of the roots of $\mathbf{b}_m$. Thus, we use the following procedure: find the roots of $\boldsymbol{\beta}_m$ and divide them into reciprocal pairs. Take the root with magnitude less than 1 from each pair and assign it to $\mathbf{b}_m$. Then form polynomial $\mathbf{b}_m$ from the roots.

To find the scale factor $\sigma_{b,m}^2$, we may equate the coefficients of the zero-th power of z for both sides of (17), resulting in

$$\sigma_{b,m}^2 \; \sum_{i=0}^{P} b_{m,i}^2 = \sigma_m^2 + \sigma_n^2 \sum_{i=0}^{P} a_i^2,$$

or

$$\sigma_{b,m}^2 = \frac{\sigma_m^2 + \sigma_n^2 \sum_{i=0}^{P} a_i^2}{\sum_{i=0}^{P} b_{m,i}^2}.$$

Thus, we have an efficient method for obtaining the ARMA filter parameters

$$\boldsymbol{\theta}_m^b = [\sigma_{b,m}^2, b_{m,1}, b_{m,2} \ldots b_{m,P}, a_1, a_2 \ldots a_P],$$

which are an equivalent set of parameters to $\boldsymbol{\theta}_m$, although they are overparameterized.

6.4 Exact Derivative Analysis of Segment PDF

We now find the exact derivatives of (15) with respect to a arbitrary parameter θ. Using standard results for matrix derivatives, we have

$$\frac{\partial \log p(\mathbf{x}_m)}{\partial \theta} = -\frac{1}{2}\text{trace}(\mathbf{C}_m^{-1}\mathbf{D}_m^{\theta}) + \frac{1}{2}\mathbf{x}_m'\mathbf{C}_m^{-1}\mathbf{D}_m^{\theta}\mathbf{C}_m^{-1}\mathbf{x}_m,$$

where $\mathbf{D}_m^{\theta}$ is the M-by-M matrix of derivatives of the elements of $\mathbf{C}_m$ with respect to θ. Since $\mathbf{C}_m$ is a symmetric Toeplitz matrix formed from the ACF sequence,

$$\mathbf{C}_m^{\theta} = \text{Toeplitz}(\mathbf{r}_m),$$

$\mathbf{D}_m^{\theta}$ is a symmetric Toeplitz matrix formed from the derivatives of the ACF sequence:

$$\mathbf{D}_m^{\theta} = \text{Toeplitz}(\mathbf{r}_m^{\theta}),$$

where

$$\mathbf{r}_m^{\theta} = [r_{0,m}^{\theta}, r_{1,m}^{\theta} \ldots r_{M-1,m}^{\theta}].$$

Finally, since the ACF is the inverse FFT of the power spectrum,

$$\mathbf{r}_m = \text{IFFT}(\rho_{0,m}^{N}, \rho_{1,m}^{N} \ldots \rho_{M-1,m}^{N}),$$

we have

$$\mathbf{r}_m^{\theta} = \text{IFFT}(\rho_{0,m}^{N\theta}, \rho_{1,m}^{N\theta} \ldots \rho_{M-1,m}^{N\theta}),$$

where $\rho_{k,m}^{N\theta}$ is the derivative of $\rho_{k,m}^{N}$ in (14) with respect to scalar parameter θ. These derivatives are given later in equations (23) through (25).

To summarize, the derivatives of (15) with respect to a parameter θ can be computed in MATLAB as

```
Cm=toeplitz(rm(1:M));
Cmi=inv(Cm);
d = real(ifft(rho_theta));
D=toeplitz(d(1:M));
lpxm_theta = -.5*trace( Cmi * D )  + .5 * x'*Cmi*D*Cmi*x;
```

where `rm` is the M-by-1 ACF vector $\mathbf{r}_m$ and `rho_theta` is $2M$-by-1 vector of derivatives $\rho_{k,m}^{2M\theta}$. This is accurate as long as the ACF corresponding to the power spectrum dies to zero at a lag less than M.

This approach is computationally of order M^3 where M may be quite large. But, since $\mathbf{C}_m$ and $\mathbf{D}_m^\theta$ are symmetric and Toeplitz, an order M^2 approach exists that employs the Levinson algorithm. This may still be prohibitive, so we seek an order M algorithm based on filtering.

6.5 Segment PDF Fisher Information Analysis - Exact

The exact second derivatives of (15) may also be obtained and the Fisher's information computed. Let θ_1, θ_2 be two arbitrary spectral parameters. Then

$$\begin{aligned}
\frac{\partial^2 \log p(\mathbf{x}_m)}{\partial\theta_1\partial\theta_2} &= \tfrac{1}{2}\text{trace}(\mathbf{C}_m^{-1}\mathbf{D}^{\theta_1}\mathbf{C}_m^{-1}\mathbf{D}^{\theta_2}) - \mathbf{x}_m'\mathbf{C}_m^{-1}\mathbf{D}^{\theta_1}\mathbf{C}_m^{-1}\mathbf{D}^{\theta_2}\mathbf{C}_m^{-1}\mathbf{x}_m \\
\mathbf{I}_m(\theta_1,\theta_2) &= -\mathcal{E}\left\{\frac{\partial^2 \log p(\mathbf{x}_m)}{\partial\theta_1\partial\theta_2}\right\} \\
&= -\tfrac{1}{2}\text{trace}(\mathbf{C}_m^{-1}\mathbf{D}^{\theta_1}\mathbf{C}_m^{-1}\mathbf{D}^{\theta_2}) + \text{trace}(\mathbf{C}_m\mathbf{C}_m^{-1}\mathbf{D}^{\theta_1}\mathbf{C}_m^{-1}\mathbf{D}^{\theta_2}\mathbf{C}_m^{-1}) \\
&= \tfrac{1}{2}\text{trace}(\mathbf{C}_m^{-1}\mathbf{D}^{\theta_1}\mathbf{C}_m^{-1}\mathbf{D}^{\theta_2})
\end{aligned}$$

The terms $\mathbf{H}^{\theta_1} = \mathbf{C}_m^{-1}\mathbf{D}^{\theta_1}$ and $\mathbf{H}^{\theta_2} = \mathbf{C}_m^{-1}\mathbf{D}^{\theta_2}$ can also be obtained efficiently using the Levinson algorithm. Despite this, the use of the above equation is primarily for validation since it is computationally expensive and not useful for combining segments.

6.6 Frequency Domain Segment PDF

In the frequency domain, it is easier to analyze how the PDF of $\mathbf{x}_m$ depends on its parameters, $\boldsymbol{\theta}$. Let $X_{k,m}^M,\ k = 0, 1 \ldots M-1$ be the DFT of $\mathbf{x}_m$

$$X_{k,m}^M = \sum_{t=1}^{M} x_t\ e^{-j2\pi k(t-1)/M}. \tag{18}$$

The frequency-domain (FD) approximation to (15) is given by the log-PDF

$$\log p(\mathbf{x}_m; \rho_{0,m}^M \dots \rho_{M-1,m}^M) = -\frac{1}{2} \sum_{k=0}^{M-1} \left\{ \log(2\pi\rho_{k,m}^M) + \frac{|X_{k,m}^M|^2}{M\rho_{k,m}^M} \right\}. \tag{19}$$

It may be verified that:

1. (19) is an approximation to (15).

2. Although written explicitly in terms of the DFT coefficients $X_{k,m}$, it is the PDF of a real multivariate Gaussian density on $\mathbf{x}_m$. That is, if it is re-written in terms of $\mathbf{x}_m$ by substituting the DFT formula (18), it may be put into the same form as (15), however the covariance matrix is not Toeplitz.

3. It is an exact PDF, that is, it integrates identically to 1 on $\mathbf{x}_m$.

4. The DFT bins are independent complex Gaussian RVs. A data sample $\mathbf{x}_m$ drawn from this PDF has independent complex Gaussian DFT bins whose expected magnitude-squared is

$$\mathcal{E}\left\{|X_{k,m}^M|^2\right\} = M\rho_{k,m}^M,\ 0 \le k < M. \tag{20}$$

PDF (19) represents a spectrally non-white Gaussian process that has independent DFT bins. It is well known that stationary Gaussian processes have DFT coefficients that are asymptotically independent as the size of the data record goes to infinity, but only truly independent if spectrally white [6]. However, because (19) is defined in terms of the magnitude-squared DFT bins, it is not a stationary process. However, it *is* a circularly stationary process.

6.7 Derivative Analysis of Segment PDF - Frequency domain

Let the first derivatives of (19) be denoted by

$$d(\theta) = \frac{\partial}{\partial\theta} \log p(\mathbf{x}_m; \boldsymbol{\theta}),$$

where θ is some arbitrary parameter upon which $\rho_{k,m}^M$ depends. Although we will not use the first derivatives of (19) themselves, their forms will help us find efficient means of finding the derivatives

of (15). We have

$$\begin{aligned} d_m(\mathbf{x}_m;\theta) \triangleq \frac{\partial}{\partial\theta}\log p(\mathbf{x}_m;\theta) &= -\tfrac{1}{2}\sum_{k=0}^{M-1}\left(\frac{\partial\rho_{k,m}^M}{\partial\theta}\right)\left\{\frac{1}{\rho_{k,m}^M}-\frac{|X_k^M|^2}{M(\rho_{k,m}^M)^2}\right\} \\ &= -\tfrac{1}{2}\sum_{k=0}^{M-1}\left(\frac{\partial\rho_{k,m}^M}{\partial\theta}\right)T_m(k), \end{aligned} \tag{21}$$

where

$$T_m(k) = \frac{1}{\rho_{k,m}^M} - \frac{|X_k^M|^2}{M(\rho_{k,m}^M)^2} \tag{22}$$

From (14), we have

$$\frac{\partial\rho_{k,m}^M}{\partial\sigma_n^2} = 1 \tag{23}$$

$$\frac{\partial\rho_{k,m}^M}{\partial\sigma_m^2} = \frac{1}{|A_k^M|^2} \tag{24}$$

$$\frac{\partial\rho_{k,m}^M}{\partial a_i} = -2\mathrm{Re}\left\{\frac{\sigma_m^2 A_k^M\, e^{j2\pi ki/M}}{|A_k^M|^4}\right\}, \quad 1\le i\le P, \tag{25}$$

leading to

$$d_m(\mathbf{x}_m;\sigma_n^2) = -\frac{1}{2}\sum_{k=0}^{M-1}\left\{\frac{1}{\rho_{k,m}^M}-\frac{|X_{k,m}^M|^2}{M(\rho_{k,m}^M)^2}\right\}, \tag{26}$$

$$d_m(\mathbf{x}_m;\sigma_m^2) = -\frac{1}{2}\sum_{k=0}^{M-1}\left\{\frac{1}{\rho_{k,m}^M}-\frac{|X_{k,m}^M|^2}{M(\rho_{k,m}^M)^2}\right\}\frac{1}{|A_k^M|^2} \tag{27}$$

$$d_m(\mathbf{x}_m;a_i) = \sum_{k=0}^{M-1}\left\{\frac{1}{\rho_{k,m}^M}-\frac{|X_{k,m}^M|^2}{M(\rho_{k,m}^M)^2}\right\}\mathrm{Re}\left\{\frac{\sigma_m^2 A_k^M\, e^{j2\pi ki/M}}{|A_k^M|^4}\right\} \quad 1\le i\le P, \tag{28}$$

We may obtain some simplification and intuitive understanding if we employ the equivalent ARMA parameterization. If we use (16) and define

$$W_{k,m}^M = \frac{X_{k,m}^M A_k^M}{B_{k,m}^M\sqrt{\sigma_{b,m}^2}}, \tag{29}$$

$$V_{k,m}^M = \frac{W_{k,m}^M}{B_{k,m}^M}, \tag{30}$$

and

$$U_{k,m}^M = \frac{W_{k,m}^M A_k^M}{B_{k,m}^M\sqrt{\sigma_{b,m}^2}}, \tag{31}$$

equations (26) through (28) may be simplified to

$$d_m(\mathbf{x}_m;\sigma_n^2) = -\frac{1}{2}\sum_{k=0}^{M-1}\left\{\frac{1}{\rho_{k,m}^M} - \frac{|U_{k,m}^M|^2}{M}\right\}, \tag{32}$$

$$d_m(\mathbf{x}_m;\sigma_m^2) = -\frac{1}{2}\sum_{k=0}^{M-1}\left\{\frac{1}{\rho_{k,m}^M|A_k^M|^2} - \frac{|V_{k,m}^M|^2}{M\sigma_{b,m}^2}\right\} \tag{33}$$

$$d_m(\mathbf{x}_m;a_i) = \frac{\sigma_m^2}{\sigma_{b,m}^2}\sum_{k=0}^{M-1}\mathrm{Re}\left\{\frac{e^{-j2\pi ik/M}}{A_k^M|B_{k,m}^M|^2}\right\} - \frac{1}{M}\sum_{k=0}^{M-1}\mathrm{Re}\left\{\frac{V_{k,m}^M\bar{V}_{k,m}^M e^{-j2\pi ik/M}}{A_k^M}\right\} \quad 1\le i\le P, \tag{34}$$

These alternative forms will help us in section 6.12.

6.8 Segment PDF Fisher Information - Frequency domain

Using (21), the Fisher's information between any two spectral parameters θ_1 and θ_2 equals

$$I_m(\theta_1,\theta_2) = -\mathcal{E}\left\{\frac{\partial^2}{\partial\theta_1\partial\theta_2}\log p(\mathbf{x}_m;\boldsymbol{\theta})\right\} = \frac{1}{2}\mathcal{E}\left\{\frac{\partial}{\partial\theta_2}\sum_{k=0}^{M-1}\left(\frac{\partial\rho_{k,m}^M}{\partial\theta_1}\right)T_m(k)\right\}$$

Before carrying out the derivative with respect to θ_2, notice that $T_m(k)$ is zero in expected value. Therefore, the only terms remaining are associated with the derivative of $T_m(k)$. Note that

$$\begin{aligned}
\mathcal{E}\left\{\tfrac{\partial}{\partial\theta_2}T_m(k)\right\} &= \mathcal{E}\left\{\left(-\tfrac{1}{(\rho_{k,m}^M)^2} + 2\tfrac{|X_{k,m}^M|^2}{M(\rho_{k,m}^M)^3}\right)\left(\tfrac{\partial\rho_{k,m}^M}{\partial\theta_2}\right)\right\} \\
&= \left(-\tfrac{1}{(\rho_{k,m}^M)^2} + 2\tfrac{M\rho_{k,m}^M}{M(\rho_{k,m}^M)^3}\right)\left(\tfrac{\partial\rho_{k,m}^M}{\partial\theta_2}\right) \\
&= \tfrac{1}{(\rho_{k,m}^M)^2}\left(\tfrac{\partial\rho_{k,m}^M}{\partial\theta_2}\right)
\end{aligned}$$

Therefore,

$$I_m(\theta_1,\theta_2) = \frac{1}{2}\sum_{k=0}^{M-1}\left(\frac{\partial\rho_{k,m}^M}{\partial\theta_1}\right)\frac{1}{(\rho_{k,m}^M)^2}\left(\frac{\partial\rho_{k,m}^M}{\partial\theta_2}\right) \tag{35}$$

Using (23) through (25), in (21) and (35), we obtain the Fisher information for the parameters $\sigma_n^2, \sigma^2, a_1, a_2 \ldots a_P$ for the segment m. We later combine them to obtain the FIM for the entire data record.

6.9 Segment PDF computation by Filtering

Efficient evaluation of (15) may be accomplished by filtering. We may write

$$\log p(\mathbf{x}_m;\boldsymbol{\theta}) = -\frac{M}{2}\log(2\pi) - \frac{1}{2}\log|\det \mathbf{C}_m| - \frac{1}{2}\mathbf{w}_m'\mathbf{w}_m, \tag{36}$$

where

$$\mathbf{w}_m = \mathbf{H}_m\,\mathbf{x}_m, \tag{37}$$

and $\mathbf{H}_m$ is the Cholesky decomposition of $\mathbf{C}_m$:

$$\mathbf{C}_m = \mathbf{H}_m'\mathbf{H}_m$$

and if $\mathbf{C}_m$ has a symmetric Toeplitz form (constant along every diagonal and consistent with any stationary process), then $\mathbf{H}_m$ is a whitening matrix that results in the covariance of $\mathbf{w}_m$ being the identity matrix. Thus, we evaluate $p(\mathbf{x}_m;\boldsymbol{\theta})$ by whitening $\mathbf{x}_m$, finding the total power, $\mathbf{w}_m'\mathbf{w}_m$, and compensating for the determinant of the whitening matrix, $|\det \mathbf{H}_m|$.

We seek an efficient filter implementation that approximates $\mathbf{H}_m$. From systems theory, we know that $\mathbf{H}_m$ has a linear shift invariant filtering equivalent. This filter is based on the power spectrum (16) which suggests an ARMA whitening filter with Z-transform

$$H_m(z) = \frac{1}{\sqrt{\sigma_{b,m}^2}}\,\frac{A(z)}{B_m(z)}. \tag{38}$$

If we begin filtering $\mathbf{x}_m$ with this filter, there will be a startup transient since the proper initial conditions are unknown. Once the transient has died out, samples at the filter output will be uncorrelated and therefore independent. We can calculate the initial samples of $\mathbf{w}_m$ exactly, then switch to the filter output after the startup transient.

6.10 Determining length of startup transient

There are many ways to measure the length of the startup transient, which is the impulse response of whitening filter $H_m(z) = A(z)/B_m(z)$. A method we have found efficient and useful is to use the FD method to solve for the autocorrelation function (ACF) corresponding to the theoretical power

spectrum of the inverse process $p_{k,m} = |A_k|^2/|B_{k,m}|^2$. To allow for an ACF up to length M, we zero-pad the polynomials $\mathbf{a}$ and $\mathbf{b}_m$ up to length $2M$, then take the magnitude-square of the the length-$2M$ DFT, followed by an inverse DFT. The result is a length-$2M$ inverse ACF estimate. In MATLAB,

```
A=fft([a(:); zeros(2*M-P-1,1)]);
B=fft([b(:); zeros(2*M-P-1,1)]);
A2=msq(A);
B2=msq(B);
ri = real(ifft(A2./B2));
```

Let T be the length of the ACF measured as the index of the last lag with amplitude larger than `ri(1)/500`:

```
K = max(find(abs(ri(1:M)) > ri(1)/500));
K=max(K,P);
```

Note that we force K to be at least as large as P, the filter order. An example of determining filter startup transient is shown in Figure 1. The MATLAB code segment below was used to produce it.

```
r = real(ifft(B2./A2));
R=toeplitz(r(1:M));
Ci=inv(chol(R))';
w1 = Ci * x(1:M);

% determine whitened samples by block of M samples
w=filter(a,b,x);
plot(w-w1);
```

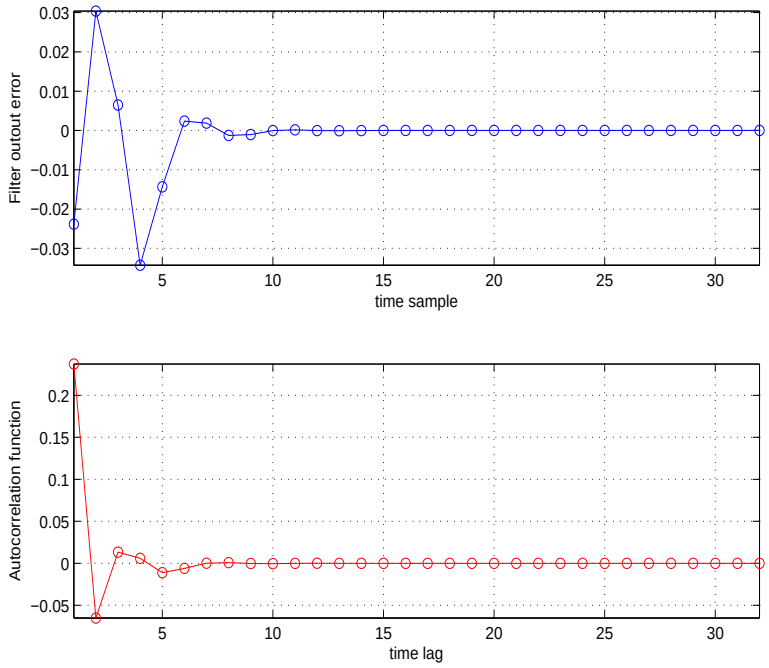

Figure 1: Example of filter startup transient. Top graph: difference between whitened samples computed in a block with samples computed by filtering. Lower graph: ACF of the inverse power spectrum. Notice that after about 10 samples, there is very close agreement, which agrees with the lower plot showing the decay of ACF corresponding to the inverse spectrum.

6.11 Extension of filter beyond K samples.

Let the length of startup transient be equal to K samples where $K \leq M$. Let us concern ourselves only with the first n samples of $\mathbf{x}$, where $K < n < M$. Modifying (36),

$$\log p(x_1, x_2 \ldots x_n; \boldsymbol{\theta}) = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\log|\det \mathbf{C}_m^n| - \frac{1}{2}\sum_{i=1}^{n} w_{i,m}^2, \tag{39}$$

where $\mathbf{C}_m^n$ is the $n \times n$ initial sub-block of $\mathbf{C}_m$. Now we ask how does this equation change as we add one more sample, x_{n+1}? Since the filter startup transient has died off, the values of $w_{i,m}$ obtained from (37) will be the same as values of w_n obtained by filtering. Therefore,

$$\log p(\mathbf{x}_m; \boldsymbol{\theta})|_{m=n+1} = \log p(x_1, x_2 \ldots x_n; \boldsymbol{\theta}) - \frac{1}{2}\log(2\pi) - \frac{1}{2}w_{n+1,m}^2 - \frac{1}{2}\log|\det \mathbf{C}_m^{n+1}/\det \mathbf{C}_m^n|,$$

however, the ratio $\det \mathbf{C}_m^{n+1}/\det \mathbf{C}_m^n$ converges rapidly to $\sigma_{b,m}^2$ as $n > K$. Thus, we have

$$\log p(x_1, x_2 \ldots x_{n+1}; \boldsymbol{\theta}) = \log p(x_1, x_2 \ldots x_n; \boldsymbol{\theta}) - \frac{1}{2}\log(2\pi\sigma_{b,m}^2) - \frac{1}{2}w_{n+1,m}^2. \tag{40}$$

Combining (40), (39), and extending out to sample M,

$$\log p(\mathbf{x}_m; \boldsymbol{\theta}) = -\frac{K}{2}\log(2\pi) - \frac{1}{2}\log|\det \mathbf{C}_m^K| - \frac{1}{2}\sum_{i=1}^{K} w_{i,m}^2 - \sum_{t=K+1}^{M}\left\{\frac{1}{2}\log(2\pi\sigma_{b,m}^2) + \frac{1}{2}w_{t,m}^2\right\}, \tag{41}$$

where $w_{i,m}$ is obtained from (37) for $i \leq K$ and from filtering for $i > K$.

6.12 Derivative Analysis of Segment PDF - filtering approach

The results of section 6.4 are still a little bit cumbersome to implement. Using filtering, we may find a much more efficient approach to finding the derivatives of PDF (15), then extend it across segment boundaries.

The filtering approach to finding the derivatives is directly analogous to section (6.11) where the segment likelihood function is formed. We begin with the exact derivatives for the first K samples, then add the filtering results to obtain the derivatives of the entire segment.

Results from the FD analysis can be used as a guide. The time-domain equivalents of equations (32) through (34) suggest the filtering approach to obtain contributions of sampes $K+1$ to M. In MATLAB notation, let

```
w=filter(a,b,x)/sqrt(sig2b);
u=filter(a,b,w)/sqrt(sig2b);
v=filter(1,b,w);
va=filter(1,a,v);
```

Then to compute the contributions of samples $K+1$ to M, equation (32) becomes

```
Dsig2n = -.5*sum(1./rho)*(M-K)/M + .5 * sum(u(K+1:M).^2);
```

Equation (33) becomes

```
Dsig2m = -.5*sum(1./rho./A2)*(M-K)/M + .5/sig2b*sum(v(K+1:M).^2);
```

Equation (34) becomes

```
da1 = real(ifft(1./conj(A)./B2));
da1=da1(2:P+1) * (M-K);
for i=1:P,
  Da(i) = sig2./sig2b * (da1(i) - sum(v(K+1:M) .* va(K+1-i:M-i)));
end;
```

7 Combining Segments for Entire data Set

We have extensively analyzed the segment PDF $p(\mathbf{x}_m;\boldsymbol{\theta}_m)$ given in (15). We have the derivatives $d_m(\mathbf{x}_m;\theta)$ for each of the parameters in $\boldsymbol{\theta}_m$ as well as FD approximations to the Fisher information matrix $\mathbf{I}_m(\boldsymbol{\theta}_m)$. We would now like to combine the results to obtain the complete data PDF $p(\mathbf{x};\boldsymbol{\Theta})$ given in (10) as well as the associated derivatives and FIM.

7.1 Combining Segment Likelihood Functions

We cannot simply add the segment PDFs to obtain the full data PDF because this would imply that the data segments are independent, and they are not. To obtain the full data PDF, we need

to extend the process of whitening which we employed within a segment in section 6.11. Note that equation (41) assumes the data spectrum (and therefore the ARMA whitening filter) remains constant, so it is not valid for samples greater than M. It is natural to ask what happens as we continue filtering when we cross over the boundary to a new segment? The only obstacle is the fact that the whitening filter changes at each segment boundary. If we have made M small enough that the filter coefficients change only slightly, we can have approximate segment-to-segment filter continuity by using the filter state variables for segment $m-1$ as the initial filter conditions for segment m, only changing the filter coefficients. We have, the main result,

$$\begin{aligned} \log p(\mathbf{x};\boldsymbol{\theta}) &= -\tfrac{K}{2}\log(2\pi) - \tfrac{1}{2}\log|\det \mathbf{C}_1^K| - \tfrac{1}{2}\sum_{t=1}^{K} w_{t,1}^2 + \sum_{t=K+1}^{M}\left\{\tfrac{1}{2}\log(2\pi\sigma_{b,1}^2) + \tfrac{1}{2}w_{t,1}^2\right\} \\ &\quad + \sum_{m=2}^{N/M}\left\{\tfrac{M}{2}\log(2\pi\sigma_{b,m}^2) + \tfrac{1}{2}\mathbf{w}_m'\mathbf{w}_m\right\} \end{aligned} \tag{42}$$

where $w_{i,m}$ is obtained from (37) for $i \le K$ and from filtering for $i > K$. The filter coefficients are re-calculated at each segment boundary and filter continuity is maintanted by retaining the filter initial conditions as the boundary is crossed.

7.2 Derivatives of the Full data PDF

One obstacle to overcome in extending the results of the segment PDF to the full PDF is the different parameterizations. In the full PDF, the power of the AR process is a scalar variance σ^2 multiplied by a time-varying envelope function $\lambda_t(\boldsymbol{\phi})$. But, in the segment PDF the signal power equals $\sigma_m^2 = L_m\sigma^2$ where L_m is the value of the piecewise constant function $\lambda_t(\boldsymbol{\phi})$ in the segment. We take a two-step approach. First we expand the parameter set to include the segment signal powers. Let the expanded parameters be denoted by

$$\boldsymbol{\Theta}' = [\sigma_n^2, a_1, a_2 \ldots a_P, \sigma_1^2 \ldots \sigma_{N/M}^2].$$

We then calculate the derivatives of $\log p(\mathbf{x};\boldsymbol{\Theta}')$ with respect to each parameter. Finally, we transform the results to obtain the derivatives with respect to parameters $\boldsymbol{\Theta}$.

In the first step, we take a leap of faith to arrive at an excellent and efficient approximation that can be validated later numerically and by comparing to the exact results. Essentially, we take the analogous approach to calculating the full data PDF by extending the filter approach in section 6.12 within a segment to multiple segments. Since we have derived a filtering implementation of the derivative calculation, we continue across segment boundaries in the same fashion - changing the filter coefficients to reflect the changed signal power yet maintaining filter continuity by retaining the filter initial conditions from the previous segment.

In the second step, we transform the results. Because L_m is a function of $\boldsymbol{\phi}$, we can write

$$\sigma_m^2 = L_m(\phi_1, \phi_2 \dots \phi_Q),$$

where Q is the dimension of $\boldsymbol{\phi}$. So, if we write $\log p(\mathbf{x}; \boldsymbol{\Theta})$ by substituting $\sigma^2 L_m(\boldsymbol{\phi})$ for σ_m^2 into $\log p(\mathbf{x}; \boldsymbol{\Theta}')$, the chain rule of derivatives gives

$$\frac{\partial \log p(\mathbf{x}; \boldsymbol{\Theta})}{\partial \sigma^2} = \sum_{m=1}^{N/M} L_m(\boldsymbol{\phi}) \left(\frac{\partial \log p(\mathbf{x}; \boldsymbol{\Theta}')}{\partial \sigma_m^2} \right),$$

$$\frac{\partial \log p(\mathbf{x}; \boldsymbol{\Theta})}{\partial \phi_i} = \sum_{m=1}^{N/M} \sigma^2 \, L_m^{\phi_i} \left(\frac{\partial \log p(\mathbf{x}; \boldsymbol{\Theta}')}{\partial \sigma_m^2} \right),$$

where

$$L_m^{\phi_i} \triangleq \frac{\partial \log L_m(\boldsymbol{\phi})}{\partial \phi_i}.$$

The remaining parameter, $\sigma_n^2, a_1, a_2 \dots a_P$ are identical so, for example

$$\frac{\partial \log p(\mathbf{x}; \boldsymbol{\Theta})}{\partial a_1} = \frac{\partial \log p(\mathbf{x}; \boldsymbol{\Theta}')}{\partial a_1}.$$

All of this can be written in the matrix form

$$\begin{bmatrix} \frac{\partial \log p(\mathbf{x};\boldsymbol{\Theta})}{\partial \sigma^2} \\ \frac{\partial \log p(\mathbf{x};\boldsymbol{\Theta})}{\phi_1} \\ \vdots \\ \frac{\partial \log p(\mathbf{x};\boldsymbol{\Theta})}{\partial \phi_Q} \\ \frac{\partial \log p(\mathbf{x};\boldsymbol{\Theta})}{\sigma_n^2} \\ \frac{\partial \log p(\mathbf{x};\boldsymbol{\Theta})}{a_1} \\ \vdots \\ \frac{\partial \log p(\mathbf{x};\boldsymbol{\Theta})}{a_P} \end{bmatrix} = \mathbf{F} \begin{bmatrix} \frac{\partial \log p(\mathbf{x};\boldsymbol{\Theta}')}{\partial \sigma_1^2} \\ \vdots \\ \frac{\partial \log p(\mathbf{x};\boldsymbol{\Theta}')}{\partial \sigma_{N/M}^2} \\ \frac{\partial \log p(\mathbf{x};\boldsymbol{\Theta}')}{\sigma_n^2} \\ \frac{\partial \log p(\mathbf{x};\boldsymbol{\Theta}')}{a_1} \\ \vdots \\ \frac{\partial \log p(\mathbf{x};\boldsymbol{\Theta}')}{a_P} \end{bmatrix},$$

where $\mathbf{F}$ is the $(P+Q+2) \times (P+1+N/M)$ matrix

$$F = \left[\begin{array}{cccc|cccc} L_1 & L_2 & \cdots & L_{N/M} & 0 & \cdots & 0 & 0 \\ \sigma^2 L_1^{\phi_1} & \sigma^2 L_2^{\phi_1} & \cdots & \sigma^2 L_{N/M}^{\phi_1} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \sigma^2 L_1^{\phi_Q} & \sigma^2 L_2^{\phi_Q} & \cdots & \sigma^2 L_{N/M}^{\phi_Q} & 0 & \cdots & 0 & 0 \\ \hline 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & 1 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 1 \end{array}\right]$$

7.3 Fisher Information of the Full data PDF

As with the derivatives, we make two steps: we obtain the FIM for the full data PDF in the segmented parameterization $\boldsymbol{\Theta}'$, then convert the results. Since accuracy of the FIM is not as critical as the accuracy of the derivatives, and because of the computational load of the FIM calculation, we settle for the FD approximations in section 6.8 simply added up over the segments. The implicit assumption here is that the segments are statistically independent, which they are not. This dependence is only an edge effect, however. Following the development above for derivatives,

we write that

$$\mathbf{I}(\mathbf{x};\boldsymbol{\Theta}) = \mathbf{F}\ \mathbf{I}(\mathbf{x};\boldsymbol{\Theta}')\ \mathbf{F}'.$$

8 Algorithm Summary

The algorithm proceeds as follows.

1. Obtain initial parameter estimates $\hat{\boldsymbol{\Theta}}$. Compute L_m, $1 \leq m \leq N/M$ according to the envelope model.

2. Determine the log-PDF, derivatives and FIM for the entire data record in terms of the extended parameter set $\boldsymbol{\Theta}'$:

   (a) Zero the accumulators for the data log-likelihood function, all derivatives, and FIM. Segment number is set to $m = 1$.

   (b) The segment power spectrum is computed for segment m (section 6.1).

   (c) Using the results of section 6.3, the ARMA filter parameters $\boldsymbol{\theta}_m^b$ for segment m are obtained.

   (d) If this is the first segment, the length of the startup transient (K) is determined (section 6.10). The whitened data for the first K samples is determined (37). Then the exact log PDF value of the first K samples is determined (equation 36). The segment data is filtered by the whitening filter (38). To form M samples of whitened data, samples 1 through K are taken from (37), while samples $K+1$ through M are taken from the filter output. The log PDF value for the M samples of the segment is calculated combining the exact PDF of the first K samples with the filter output (equation 41). Add the segment log PDF value to the log-PDF accumulator. Save the filter initial conditions.

   (e) If this is not the first segment, the segment data is filtered by the whitening filter (38) using the stored initial conditions from the previous segment. This is used to compute

the segment log-PDF (see the last term in (42). Add the segment log PDF value to the log-PDF accumulator. Save the filter initial conditions.

(f) If this is the first segment, use section (6.4) to obtain the exact derivatives for a block of the first K samples. Then, use section (6.12) to obtain the contribution of samples $K+1$ through M by filtering. Save all initial conditions for the various filters. Add the segment contribution to the derivative accumulators. Note that the derivative with respect to parameter σ_m^2 gets a contribution only for segment m.

(g) Use section (6.8) to obtain the FD approximation to the segment FIM. Add to the FIM accumulator. Note that terms involving parameter σ_m^2 get a contribution only for segment m.

(h) Repeat steps (a) through (h) until all segments have been processed. At this point, we have the log-PDF value for the entire data record as well as derivatives of the log-PDF for each parameter in $\boldsymbol{\Theta}'$.

3. Convert the derivatives and FIM that are in terms of $\boldsymbol{\Theta}'$ to the parameters set $\boldsymbol{\Theta}$ (section 7.2).

4. Update the parameter $\hat{\boldsymbol{\Theta}}$ estimates using (8).

5. Repeat steps 2 through 4 until $\hat{\boldsymbol{\Theta}}$ converges. Monitor the log-likelihood function, it should increase at each step or remain the same.

9 Simulation

We conducted two experiments. In the first, a simple smaller experiment was used as a means of comparing the exact PDF (10) and numerically computed derivatives with the FD approach (sections 6.6 and 6.7) and filtering approach (sections 6.11 and 6.12). We used a Gaussian shaped envelope function:

$$L_m = (2\pi V)^{-1/2} \ \exp^{-(m-\mu)^2/(2V)},$$

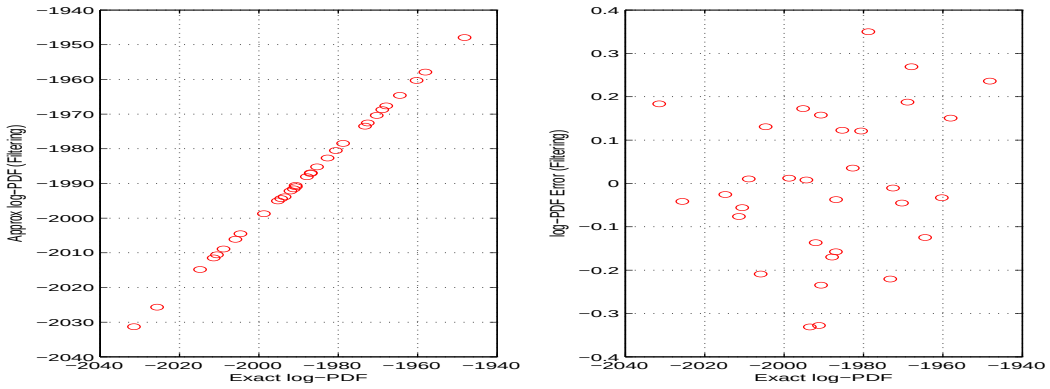

Figure 2: Comparison of log-PDF values for 24 trials using filtering approximation. Left panel: approximate vs. exact log-PDF. Right panel: approximation error vs. exact log-PDF.

where μ, V are the mean and variance of the Gaussian shape. The parameters we used are: $M = 48$ samples, $N/M = 16$ segments, $\sigma^2 = 60$, $\sigma_n^2 = 5$, $V = 28.44$, $\mu = 8.5$, and an order-2 AR model with

```
a=[1.0000    -0.9899     0.4900]
```

The envelope function λ_t was a step-wise constant function equal to L_m in each segment. For each of 24 trials, we computed the exact log-PDF according (9) and (10). Next, we computed the log-PDF according to the algorithm described in section 8 which uses the filtering approximation (equation 42). In figure 2 we compare the exact PDF values with the approximation. There is very close agreement with log-PDF error within +/-0.4. The same experiment was conducted using the FD approximation to the segment PDF (section 6.6), accumulated over the segments. The results are shown in figure 3 showing a bias of -4 and a variation of +/-10. This clearly shows the superiority of the filtering approach as compared with the FD approach. A similar comparison can be made for the derivatives of the log-PDF. In figures 4 and 5, we see very close agreement with the exact value for the algorithm of section 8 (which uses the filtering approach from section 7.2), whereas the FD derivatives have significant errors, especially the AR parameters. If we look more closely at the derivatives for parameter a_1 (figure 6), we see a dramatic improvement when the filtering approach is used.

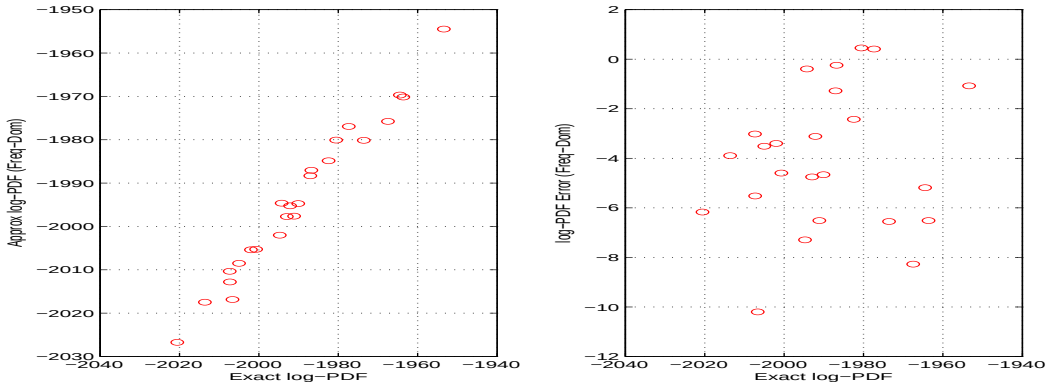

Figure 3: Comparison of log-PDF values for 24 trials using FD approximation. Left panel: approximate vs. exact log-PDF. Right panel: approximation error vs. exact log-PDF.

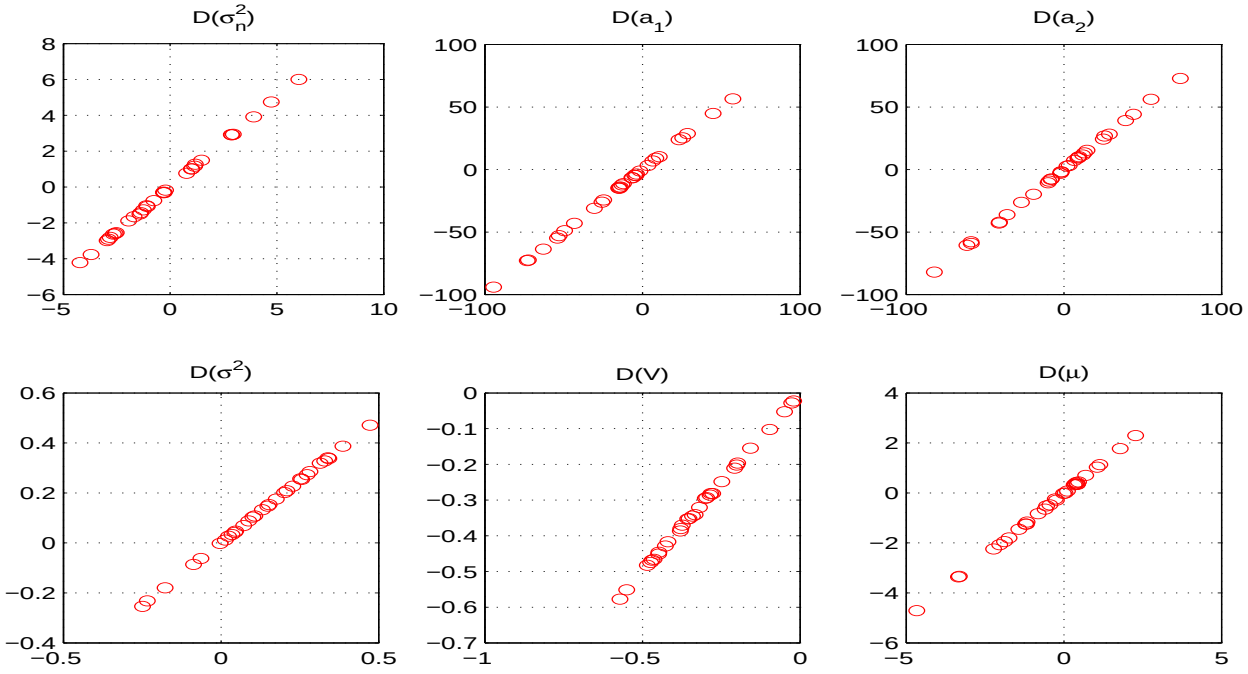

Figure 4: Comparison of log-PDF derivatives values for 24 trials using filtering approximation. In each panel the exact derivative, determined numerically from equation (10), is plotted on the X axis and the approximation is plotted on the Y axis.

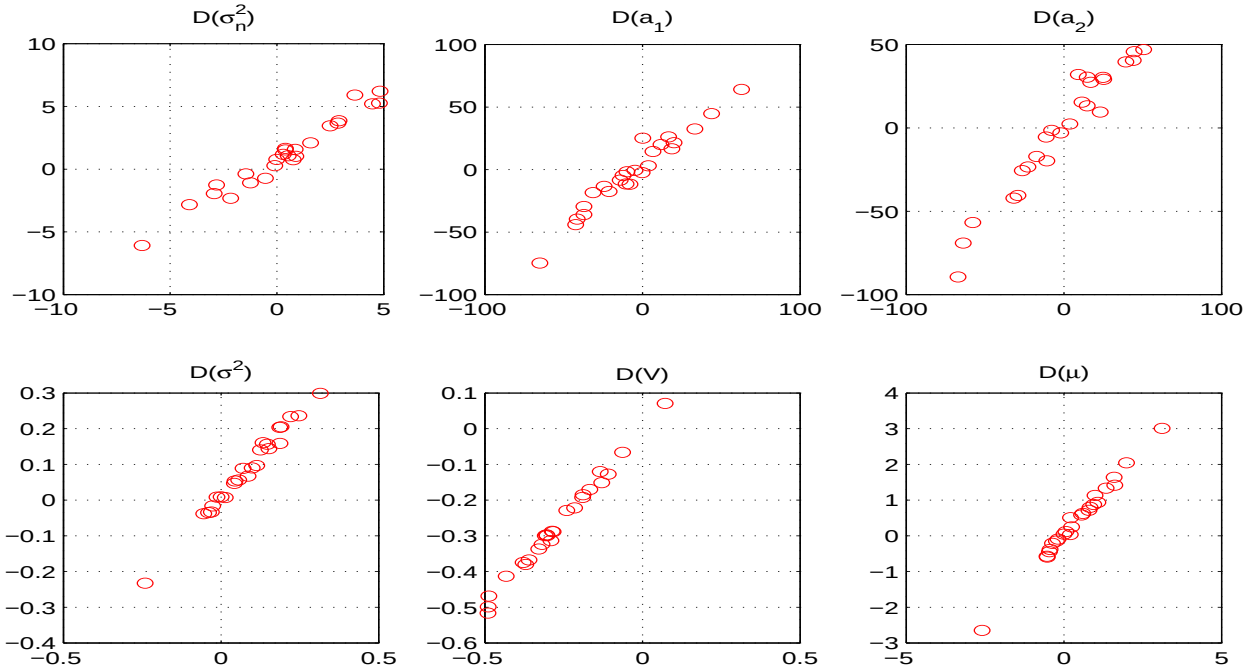

Figure 5: Comparison of log-PDF derivatives values for 24 trials using FD approximation. In each panel the exact derivative, determined numerically from equation (10), is plotted on the X axis and the approximation is plotted on the Y axis.

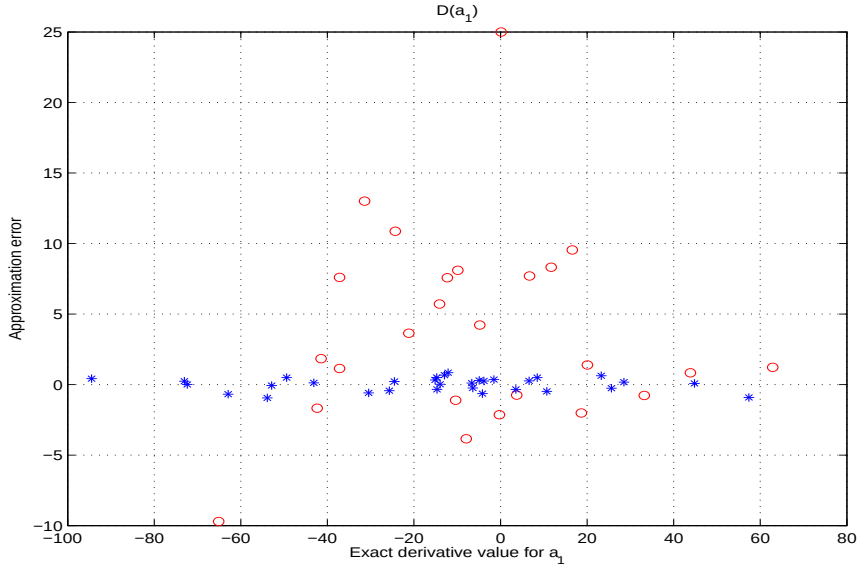

Figure 6: Log-PDF derivative errors for FD method (circles) and filtering method (asterix).

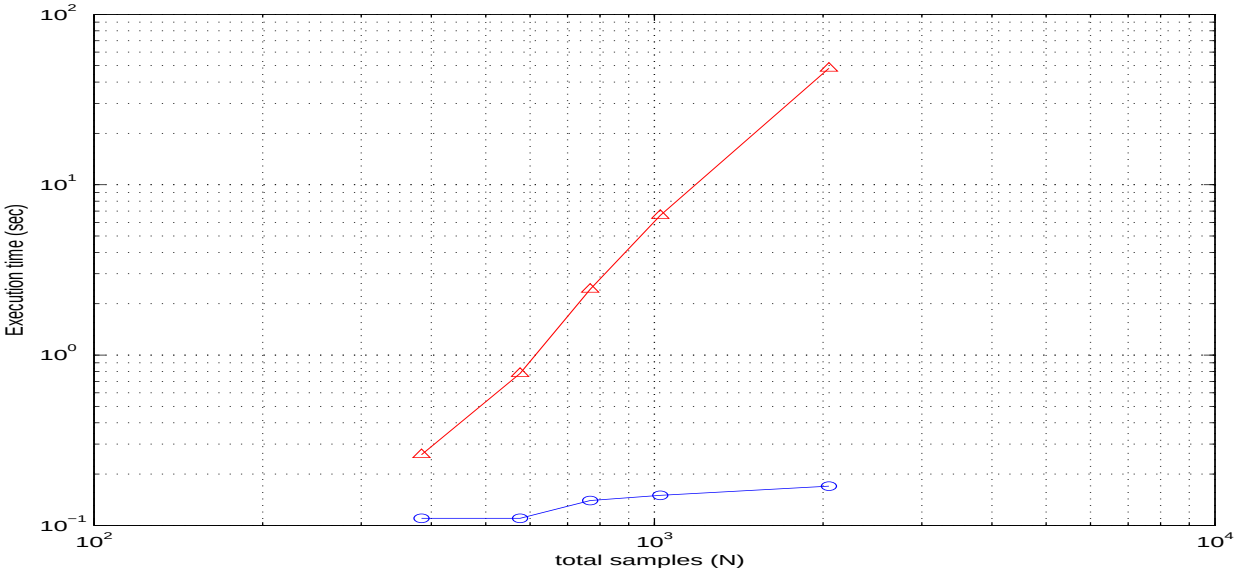

Figure 7: Comparison of execution times for equation (10) (triangles) and the procedure in section 8 (circles) as a function of N.

To compare execution times, we compare the time to execute the procedure in section 8 with equation (10) as a function of the total number of samples N. The results are shown in figure 7.

In the second experiment, we demonstrated the parameter estimation accuracy. The parameters we used are: $M = 128$ samples, $N/M = 60$ segments, $\sigma^2 = 60$, $\sigma_n^2 = 5$, $V = 56.2$, $\mu = 30.5$, and an order-4 AR model with

```
a=[1.0000    -1.2124     1.2125    -0.8760     0.3540]
```

An example of simulated data is shown in Figure 8. To demonstrate the whitening process, the spectrogram of the concatenated whitened samples $w_{t,i}$, $1 \leq i \leq N/M$, $1 \leq t \leq M$ is also shown. The true values of the parameters were not used in the simulation except to create the data. Initial parameter estimates were obtained by an ad-hoc means, then used as a starting point in the algorithm. A typical maximum-likelihood convergence cycle is as follows:

```
lpX(1)=-18773.774317, del=9058.229974, step=0.500000
lpX(2)=-18294.994823, del=478.779494, step=1.000000
```

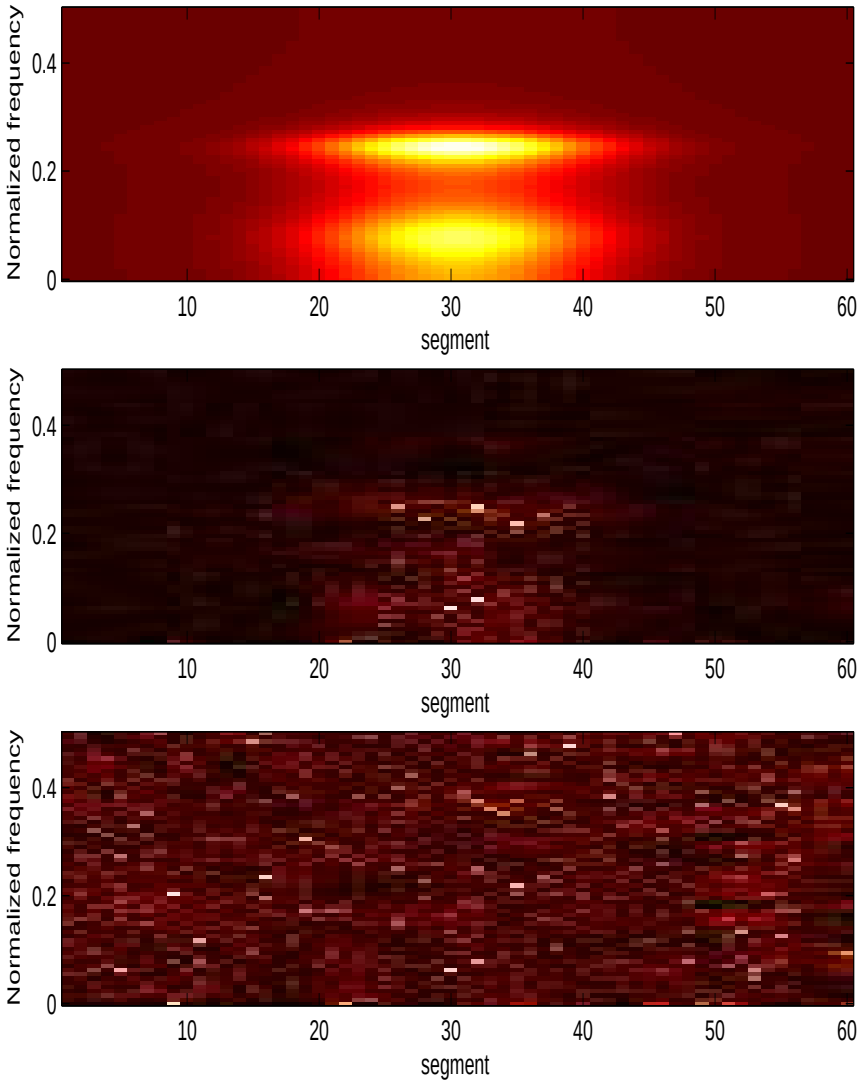

Figure 8: Example of simulated data. Top frame: theoretical power spectrum as a function of segment. Center panel: example of spectrogram of data. Bottom panel: spectrogram of whitened data $w_{t,i}$ obtained from time-varying ARMA filter in accordance with true parameter values.

```
lpX(3)=-18290.429420, del=4.565404, step=1.000000
lpX(4)=-18274.643158, del=15.786262, step=1.000000
lpX(5)=-18274.241121, del=0.402037, step=1.000000
lpX(6)=-18274.239577, del=0.001544, step=1.000000
lpX(7)=-18274.239531, del=0.000046, step=1.000000
lpX(8)=-18274.239530, del=0.000001, step=1.000000
lpX(9)=-18274.239530, del=0.000000, step=1.000000
```

where `lpX(i)` is the log-PDF value for iteration `i`, `del` is the amount of increase in `lpX`, and `step` is the step size a multiplicative factor that is normally 1. Notice the rapid convergence, decreasing `del` by over an order of magnitude per step. This is a sign that the derivatives as well as FIM are correct.

A better indication that the algorithm is working can be had by comparing simulated parameter covariance with the CR bound. We created 1000 examples of the data record. On each record, we iterated to obtain the ML parameter estimates. The parameters $\mathbf{\Theta}$ had dimension 8 and consisted of

$$\mathbf{\Theta} = [\sigma_n^2, a_1, a_2, a_3, a_4, \sigma^2, V, \mu].$$

The average parameter estimates of 1000 trials are

```
True parameter values
   [5.0000 -1.2124  1.2125 -0.8760  0.3540 60.0000 56.2000 30.5000]
Average of 1000 trials:
   [5.0033 -1.2097  1.2081 -0.8734  0.3518 59.7153 56.5399 30.5487]
```

The mean and covariance of the estimates are shown below along with the true values and the CR bound. Note that unlike the likelihood function and its first derivatives, the FIM (CR bound) does not depend on the data dircetly; it depends only on the parameters. If the true parameter values are substituted in, the result is the theoretical FIM and CR bound, although we used the

FD approximation. In figure 9, we compare the empirical parameter estimation error covariance with the inverse of the theoretical FIM. The match is quite good. One half of the log determinant of the FIM is a component of the denominator of the J-function. Note that the two matrices above have half-log-determinats of -15.55 and -15.12, respectively.

10 Conclusion

We have presented a model for autoregressive processes with time-varying amplitude in white Gaussian noise. Because the exact theoretical implemetation of the probability density function (PDF) is cumbersome, we have derived a very accurate and efficient filter-based implementation for use in a maximum likelihood (ML) framework or in a class-specific classifier. The key simplifying assumption is that the amplitude varies relatively slowly so that for a given time, M samples, the process can be regarded as fixed and a linear shift-invariant filter can whiten the data. The model uses the most compact parameterization and the likelihood function is computed very efficiently using filtering. At each M-sample segment, the whitening filter is recalculated and the filter continues. The filtering approach is extended to the calculation of the log-likelihood derivatives which are essential in order to iterate to obtain the ML estimates. We demonstrated maximum likelihood estimation on simulated data. Results obtained using an efficient filtering method are compared with the exact formulas and show not only very close agreement but orders of magnitude lower processing requirements. As a final check, empirical parameter estimation covariance is compared with and agrees closely with the Cramer-Rao (CR) lower bound.

References

[1] S. Kay, *Modern Spectral Estimation: Theory and Applications.* Prentice Hall, 1988.

[2] W. J. Done, "Estimation of the parameters of an autoregressive process in the presence of additive white noise," *NASA STI/Recon Technical Report N*, vol. 79, pp. 27363–+, Dec. 1978.

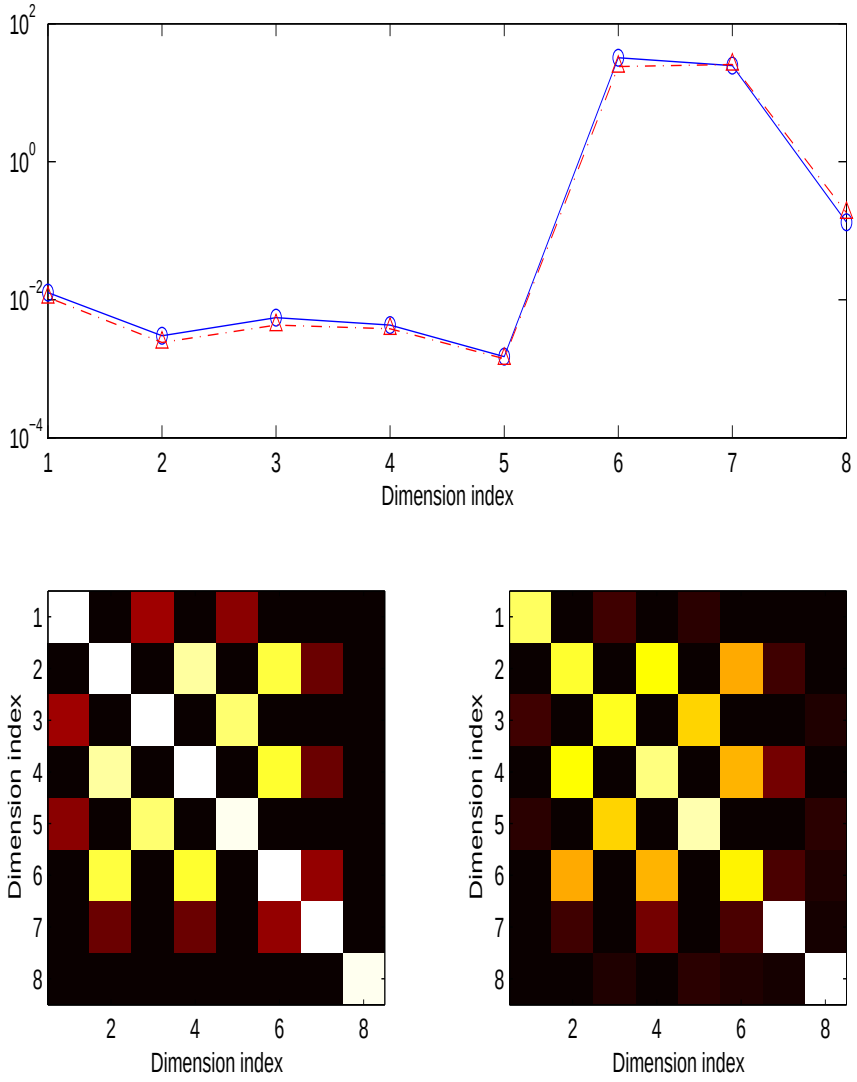

Figure 9: Comparison of empirical parameter estimation error covariance matrix with the CR bound (inverse of the average FIM). Top graph: comparison of diagonal elements. Solid line: empirical covariance, dotted line: CR bound. Bottom panels: CR bound and empirical covariance matrices normalized and displayed as an image. Both the CR bound matrix and the empirical covariance were normalized by a diagonal similarity transformation to which resulted in a constant diagonal for the CR bound matrix.

[3] W. G. Chung, "Iterative Autoregressive Parameter Estimation in presence of additive white noise," *Electronic Letters*, vol. 27, pp. 1800–1802, Sept. 1991.

[4] P. M. Baggenstoss, "The PDF projection theorem and the class-specific method," *IEEE Trans Signal Processing*, pp. 672–685, March 2003.

[5] H. Wold, *A Study in Analysis of Stationary Time Series*. Uppsala: Almqvist und Wiksel, 1938.

[6] T. W. Anderson, *The Statistical Analysis of Time Series*. Wiley, 1971.