Optimum Block Modified Covariance Algorithms For Spectral Analysis

Kyriakos Kitsios*, Andreas Spanias*, Bruno Welfert**

*Telecommunications Research Center
Department of Electrical Engineering
Arizona State University, Tempe, AZ 85287, USA
**Department of Mathematics, ASU, Tempe, AZ 85287, USA

Abstract— **Optimum Block Modified Covariance Algorithms are developed for the computation of time-varying autoregressive (AR) parameters. The proposed algorithms are gradient-based and minimize simultaneously the forward and backward linear prediction error in the block least squares sense. The methods presented here differ from those presented previously [1] in that they use optimally selected time-varying convergence factors such that the block mean square error is minimized from one iteration to the next. In particular, one of the algorithms developed, which is called the Block Modified Covariance Algorithm with individual adaptation of parameters (BMCAI), uses individual time-varying convergence factors computed using modified covariance matrix approximations along with the Gauss-Seidel method. It is shown that the BMCAI can be used efficiently for time-varying spectral estimation and performs as well as any of the fixed-window least squares modified covariance algorithms [2].**

I. Introduction

This paper is concerned with the development of efficient algorithms for least-squares forward-backward prediction (FBP). The unconstrained minimization of the FBP error was originally proposed by Ulrych and Clayton [3] and Nuttall [4] and is known as the modified covariance (MC) method. The name modified covariance stems from the fact that the range over which the prediction errors are minimized is the same as that of the non-windowed method of linear prediction. Algorithms minimizing the unconstrained FBP error, as opposed to those using the Levinson recursion (e.g., Burg [5], etc), do not suffer from spectral line splitting and became very popular in the early 1980s. Unconstrained FBP requires matrix inversion and most of the originally proposed algorithms compute AR parameters based on a fixed-window approach. Marple developed a fast Cholesky algorithm (FCA) [2] which requires $O(p^2)$ operations and more recently a fast QR algorithm (FQRA) [6] which was shown to have improved numerical behavior relative to the FCA. The fast inversion [2], [6] algorithms are order recursive and operate on a fixed N-point record, i.e., they are non-adaptive. A fixed-order sliding window QR method for the linear prediction problem was proposed in [7]. Other fast recursive algorithms for FBP were proposed for the prewindowed and non-windowed cases by Kalouptsidis and Theodoridis [8] and Berberidis and Theodoridis [9] respectively. A family of fixed-order sliding-window block gradient algorithms for FBP, namely the block modified covariance algorithms (BMCAs), were proposed recently by Spanias [1]. In particular, the BMCA worked reasonably well in a series of "benchmark" simulations, however its performance deteriorated considerably in scenaria requiring estimation of the spectral content of multiple closely-spaced sinusoids. This is mainly because all the block modified covariance algorithms (BMCAs) use a single convergence factor (or step size μ_B) which does not allow for fast adaptation in cases where the modified covariance matrix has high eigenvalue disparity. In this paper, we concentrate on the development of multiple convergence factors for adapting the AR parameters. The use multiple convergence factors in adaptive FBP was motivated by work done in adaptive FIR system identification by Mikhael *et al* [10]. The difference between the algorithms presented in this paper and those presented by Mikhael are: a) the algorithms presented here are intended for modified covariance linear prediction in which the structure of the equations to be solved is distinctly different than that encountered in FIR system identification, b) the algorithms presented are studied in the context of spectral estimation applications and deal with the idiosyncrasies of some of complex spectral estimation examples such as multiple closely spaced sinusoids, and c) the proposed methods go a step beyond Mikhael's work in the sense that the computation of the individual μ_B is done efficiently using fast and stable Gauss-Seidel numerical methods tailored specifically to deal with the structure of the modified covariance equations. The latter is the most important contribution of the paper in that it provides opportunities for reducing the complexity of the algorithms by using

approximates of the modified covariance matrix while maintaining the attractive performance characteristics of least squares MC spectral estimators.

The rest of the paper is organized as follows. In section II, the BMCA is formulated and time-varying convergence factors are developed for: a) scaling the combined forward-backward gradient (BMCA-2) and b) scaling separately the forward and backward gradients (BMCA-3). Section III describes an algorithm that uses individual step sizes for adapting the AR parameters (BMCAI). An efficient Gauss-Seidel iterative procedure for computing the optimum convergence factors is also presented in this section. Section IV, presents several simulations using the BMCAI and Section V gives the conclusions.

II. The Block Time-Domain MC Algorithm

A. *Problem formulation*

In this section, a general technique for formulating the BMCAI is presented. We begin by defining the following parameters: let i be the block index, p the order of the AR model, N the number of samples for prediction, $2L$ the length of the processed block, n the time index, $a_k(i)$ the k-th adjustable parameter in the i-th block ($k = 1, 2, 3, ..., p$), $x(n)$ the input signal for linear prediction (adaptive filter), $e_\ell(i)$ the ℓ-th error signal in the i-th block ($\ell = 1, 2, ..., 2L$), and S the number of samples per block shift.

At the i-th iteration, the objective is to minimize the cost function $J(i+1) = \frac{1}{2}\boldsymbol{e}_{fb}^T(i+1)\boldsymbol{e}_{fb}(i+1)$ where the $2L \times 1$ error vector $\boldsymbol{e}_{fb}(i)$ is given by

$$\begin{aligned}\boldsymbol{e}_{fb}(i) &= [e_f(iS+p+1)...e_f(iS+N) \\ &\quad e_b(iS+p+1)...e_b(iS+N)]^T \end{aligned} \quad (1)$$

and $e_f(n)$ and $e_b(n)$ are the forward and backward prediction errors

$$e_f(n) = x(n) - \sum_{k=1}^{p} a_k(i)x(n-k), \quad (2)$$

$$e_b(n) = x(n-p) - \sum_{k=1}^{p} a_k(i)x(n-p+k). \quad (3)$$

Equations (1), (2) and (3) can be written block-wise as

$$\boldsymbol{e}_{fb}(i) = \boldsymbol{x}(i) - \boldsymbol{X}_{fb}(i)\boldsymbol{a}(i) \quad (4)$$

where the $2L \times 1$ vector $\boldsymbol{x}(i)$ is given by

$$\begin{aligned}\boldsymbol{x}(i) &= [x_f(iS+p+1).....x_f(iS+N) \\ &\quad x_b(iS+1).....x_b(iS+N-p)]^T \end{aligned} \quad (5)$$

and the $2L \times p$ matrix $\boldsymbol{X}_{fb}(i)$ and $p \times 1$ vector $\boldsymbol{a}(i)$ are defined by

$$\boldsymbol{X}_{fb}(i) = \begin{bmatrix} x(iS+p) & ... & x(iS+1) \\ x(iS+p+1) & ... & x(iS+2) \\ . & . & . \\ x(iS+N-1) & ... & x(iS+N-p) \\ x(iS+2) & ... & x(iS+p+1) \\ x(iS+3) & ... & x(iS+p+2) \\ . & . & . \\ x(iS+N-p+1) & ... & x(iS+N) \end{bmatrix}$$

$$\boldsymbol{a}(i) = [a_1(i), a_2(i), ..., a_p(i)]^T. \quad (6)$$

The BMCA uses the method of steepest descent and the parameter update formula can be written in a vector form as $\boldsymbol{a}(i+1) = \boldsymbol{a}(i) - \mu\nabla_{fb}(i)$, with $\nabla_{fb}(i) = -\frac{2}{2L}\boldsymbol{X}_{fb}^T(i)\boldsymbol{e}_{fb}(i)$. A necessary condition for the non-singularity of the covariance matrix $\boldsymbol{R}(i) = \boldsymbol{X}_{fb}^T(i)\boldsymbol{X}_{fb}(i)$ is that the block size $2L$ be greater than the order of linear prediction p. The condition for convergence of the algorithm is $0 < \mu < 2L/\lambda_{max}$, where λ_{max} is the largest eigenvalue of $E(\boldsymbol{X}_{fb}^T(i)\boldsymbol{X}_{fb}(i))$.

B. *Optimum convergence factor*

The optimum convergence factor for the BMCA-2 is derived in [1] by minimizing $J(i+1)$ w.r.t. $\mu_{fb}(i)$. This is done in [1] by expressing the error at the $(i+1)$-st iteration as a function of the error at the i-th iteration using a truncated Taylor series approximation, i.e.,

$$\begin{aligned}\boldsymbol{e}_{fb}(i+1) &= \boldsymbol{e}_{fb}(i) + \frac{\partial \boldsymbol{e}(i)}{\partial \boldsymbol{a}(i)}\Delta\boldsymbol{a}(i) \\ &= \boldsymbol{e}_{fb}(i) - \boldsymbol{X}_{fb}(i)(\boldsymbol{a}(i+1) - \boldsymbol{a}(i)) \\ &= \boldsymbol{e}_{fb}(i) + \mu_{fb}(i)\boldsymbol{X}_{fb}(i)\nabla_{fb}(i) \end{aligned} \quad (7)$$

and solving the quadratic minimization problem $\min_{\mu(i)} J(i+1)$. Its solution is

$$\mu_{fb}(i) = \frac{\nabla_{fb}^T(i)\nabla_{fb}(i)}{\nabla_{fb}^T(i)\boldsymbol{X}_{fb}^T(i)\boldsymbol{X}_{fb}(i)\nabla_{fb}(i)}. \quad (8)$$

This optimal value may be multiplied by a damping factor $\beta \leq 1$ in order to improve the misadjustment of the BMCA-2. The convergence speed of the algorithm improves when the time varying $\mu_{fb}(i)$ is used in a non-stationary environment because it adapts to the local statistics of the processed block [1].

C. *Separate convergence factors for forward-backward prediction*

Let the $L \times p$ matrices $\boldsymbol{X}_f$ and $\boldsymbol{X}_b$ contain the input data for forward and backward linear predictions, respectively. Since $\boldsymbol{X}_{fb}(i) = \begin{bmatrix} \boldsymbol{X}_f(i) \\ \boldsymbol{X}_b(i) \end{bmatrix}$ and $\boldsymbol{e}_{fb}(i) =$

$[e_f(i)e_r(i)]^T$ we obtain

$$\nabla_{fb}(i) = \nabla_f(i) + \nabla_b(i), \tag{9}$$

i.e., the forward-backward gradient $\nabla_{fb}(i)$ is decomposed into a forward gradient $\nabla_f(i) = -\frac{2}{2L}\boldsymbol{X}_f^T(i)\boldsymbol{e}_f(i)$ and a backward gradient $\nabla_b(i) = -\frac{2}{2L}\boldsymbol{X}_b^T(i)\boldsymbol{e}_b(i)$. Thus we now consider two different convergence factors μ_f and μ_b in the update

$$\boldsymbol{a}(i+1) = \boldsymbol{a}(i) - \mu_f(i)\nabla_f(i) - \mu_b(i)\nabla_b(i). \tag{10}$$

The idea is again to optimize both time varying convergence factors $\mu_f(i)$ and $\mu_b(i)$ in the present block such that the total combined forward-backward error in the next block (i.e., $J(i+1)$) is minimized. Following the same procedure as before we get

$$\boldsymbol{e}_{fb}(i+1) = \boldsymbol{e}_{fb}(i) + \mu_f\boldsymbol{X}_{fb}\nabla_f + \mu_b\boldsymbol{X}_{fb}\nabla_b. \tag{11}$$

The conditions

$$\frac{\partial J(i+1)}{\partial \mu_f(i)} = \frac{\partial J(i+1)}{\partial \mu_b(i)} = 0$$

lead to the 2×2 linear symmetric system of equations

$$\begin{bmatrix} \boldsymbol{d}_f^T(i)\boldsymbol{d}_f(i) & \boldsymbol{d}_f^T(i)\boldsymbol{d}_b(i) \\ \boldsymbol{d}_f^T(i)\boldsymbol{d}_b(i) & \boldsymbol{d}_b^T(i)\boldsymbol{d}_b(i) \end{bmatrix} \begin{bmatrix} \mu_f(i) \\ \mu_b(i) \end{bmatrix} = \begin{bmatrix} \nabla_{fb}^T(i)\nabla_f(i) \\ \nabla_{fb}^T(i)\nabla_b(i) \end{bmatrix} \tag{12}$$

where $\boldsymbol{d}_f(i) = \boldsymbol{X}_{fb}(i)\nabla_f(i)$ and $\boldsymbol{d}_b(i) = \boldsymbol{X}_{fb}(i)\nabla_b(i)$. This algorithm is designated as BMCA-3.

III. BMCA With Individual Adaptation Of Parameters

The concept of using individual constant or time-varying convergence factors has been examined in [10]. The regular gradient type adaptive algorithms use one constant or time-varying step size which is chosen to be the same for all the filter parameters. We propose here to use individual convergence factors which are optimally chosen to adapt *individual filter parameters*. The values of the step sizes are, as before, updated at each block iteration.

A. *Problem formulation*

We now consider the relation

$$\boldsymbol{a}(i+1) = \boldsymbol{a}(i) - \boldsymbol{M}(i)\nabla_{fb}(i) \tag{13}$$

to update the parameters, where $\boldsymbol{M}(i)$ is a $p \times p$ diagonal matrix containing the p convergence factors, i.e.,

$$\boldsymbol{M}(i) = \begin{bmatrix} \mu_1(i) & & \\ & \ddots & \\ & & \mu_p(i) \end{bmatrix}. \tag{14}$$

Because it is difficult to compute an ensemble average the block gradient vector $\nabla_{fb}(i)$ is replaced by an estimated block gradient vector which is given by

$$\widehat{\nabla}_{fb}(i) = \frac{1}{L}\frac{\partial J(i)}{\partial \boldsymbol{a}(i)} = -\frac{1}{L}\boldsymbol{X}_{fb}^T(i)\boldsymbol{e}_{fb}(i). \tag{15}$$

From (13), (14), (15) one obtains the general form of the parameter updating formula in matrix vector form as:

$$\boldsymbol{a}(i+1) = \boldsymbol{a}(i) + \frac{1}{L}\boldsymbol{M}(i)\boldsymbol{X}_{fb}^T(i)\boldsymbol{e}_{fb}(i). \tag{16}$$

In the parameter update (16), there are p individual time-varying convergence factors, $\mu_k(i)$ $(k = 1, 2, ..., p)$. These factors are chosen at each iteration i so as to minimize the functional $J(i+1)$. To this end, the forward and backward errors are expanded using the truncated Taylor series

$$\begin{aligned} \boldsymbol{e}_{fb}(i+1) &= \boldsymbol{e}_{fb}(i) + \frac{\partial \boldsymbol{e}_{fb}(i)}{\partial \boldsymbol{a}(i)}\Delta\boldsymbol{a}(i) \\ &= \boldsymbol{e}_{fb}(i) - \boldsymbol{X}_{fb}(i)(\boldsymbol{a}(i+1) - \boldsymbol{a}(i)) \\ &= \boldsymbol{e}_{fb}(i) - \frac{1}{L}\boldsymbol{X}_{fb}(i)\boldsymbol{M}(i)\boldsymbol{X}_{fb}^T(i)\boldsymbol{e}_{fb}(i) \\ &= \boldsymbol{e}_{fb}(i) - \boldsymbol{X}_{fb}(i)\boldsymbol{M}(i)\boldsymbol{q}(i) \end{aligned} \tag{17}$$

with $\boldsymbol{q}(i) = \frac{1}{L}\boldsymbol{X}_{fb}^T(i)\boldsymbol{e}_{fb}(i) = -\nabla_{fb}(i)$. Here the partial derivative $\frac{\partial \boldsymbol{e}_{fb}(i)}{\partial \boldsymbol{a}(i)}$ is obtained from (4) and reduces to $-\boldsymbol{X}_{fb}(i)$.

The next step is to choose $\boldsymbol{M}(i)$ such that $J(i+1)$ is minimized. This is done by setting

$$\frac{\partial J(i+1)}{\partial \mu_k(i)} = 0 \tag{18}$$

for $k = 1, ..., p$. This leads to the system of equations

$$\boldsymbol{q}^T(i)\boldsymbol{M}(i)\boldsymbol{R}(i)\frac{\partial \boldsymbol{M}(i)}{\partial \mu_k(i)}\boldsymbol{q}(i) = \boldsymbol{q}^T(i)\frac{\partial \boldsymbol{M}(i)}{\partial \mu_k(i)}\boldsymbol{q}(i) \tag{19}$$

for $k = 1, ..., p$, or

$$\boldsymbol{R}(i)\boldsymbol{M}(i)\boldsymbol{q}(i) = \boldsymbol{q}(i). \tag{20}$$

Equivalently,

$$\boldsymbol{M}(i)\boldsymbol{q}(i) = \boldsymbol{R}^{-1}(i)\boldsymbol{q}(i), \tag{21}$$

so that the updating formula (13) becomes

$$\boldsymbol{a}(i+1) = \boldsymbol{a}(i) + \boldsymbol{M}(i)\boldsymbol{q}(i) = \boldsymbol{a}(i) + \boldsymbol{R}^{-1}(i)\boldsymbol{q}(i). \tag{22}$$

The last equation is the weight update equation for the BMCAI with individual adaptation of parameters. Its main drawback is the requirement of computing the solution of a system of equations of order p. The associated cost can become intolerable especially for high-order adaptive system. The following sections contain various approaches which can be used to approximate the quantity $\boldsymbol{R}^{-1}(i)\boldsymbol{q}(i)$ in an efficient manner.

B. Implementation via the Matrix Inversion Lemma

As in the OBAI algorithm [10] and the GOBA algorithm described in [11], the BMCAI with individual adaptation of parameters requires the determination of the solution of a $p \times p$ symmetric and generally positive definite system $\boldsymbol{R}(i)\boldsymbol{z}(i) = \boldsymbol{q}(i)$. A common approach to reduce the cost of the solution process consists in updating $\boldsymbol{R}^{-1}(i)$ as well by taking advantage of the matrix inversion lemma. Although overlaps of $L - S$ signals in consecutive blocks can be handled this way for any value of S, considering $S = 1$ both simplifies the exposition and the computational cost as well, for $\boldsymbol{X}(i+1)$ then "resembles" more to $\boldsymbol{X}(i)$. In the remainder of the paper we thus suppose that $S = 1$.

Using the partitioning of $\boldsymbol{X}_{fb}(i)$ introduced in Section II C., we obtain

$$\begin{aligned} \boldsymbol{R}(i) &= \boldsymbol{X}_{fb}^T(i)\boldsymbol{X}_{fb} = \boldsymbol{X}_f^T(i)\boldsymbol{X}_f + \boldsymbol{X}_b^T(i)\boldsymbol{X}_b \\ &= \boldsymbol{R}_f(i) + \boldsymbol{R}_b(i). \end{aligned} \tag{23}$$

where $\boldsymbol{R}_f$ is the covariance matrix based on the forward prediction and $\boldsymbol{R}_b$ is the covariance matrix based on the backward prediction. If $\boldsymbol{x}_\ell(i)$ denotes the ℓ-th row of $\boldsymbol{X}_{fb}(i)$ ($\ell = 1, ..., 2L = 2(N-p)$), the following recurrence relation for $\boldsymbol{X}_b(i)$ then holds:

$$\boldsymbol{X}_b(i+1) = \boldsymbol{U}\boldsymbol{X}_b(i) + \boldsymbol{u}\boldsymbol{x}_{2L}(i+1). \tag{24}$$

Here

$$\boldsymbol{U} = \begin{bmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & 0 \end{bmatrix} \quad \text{and} \quad \boldsymbol{u} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}.$$

Because $\boldsymbol{X}_f(i+1) = \boldsymbol{X}_b(i)\boldsymbol{P}$ with

$$\boldsymbol{P} = \begin{bmatrix} & & 1 \\ & \iddots & \\ 1 & & \end{bmatrix},$$

$\boldsymbol{X}_f(i+1)$ also satisfies a recurrence relation similar to (24). We obtain

$$\begin{aligned} \boldsymbol{R}_b(i+1) &= \boldsymbol{X}_b^T(i+1)\boldsymbol{X}_b(i+1) \\ &= \left(\boldsymbol{X}_b^T(i)\boldsymbol{U}^T + \boldsymbol{x}_{2L}^T(i+1)\boldsymbol{u}^T\right) \\ &\quad \left(\boldsymbol{U}\boldsymbol{X}_b(i) + \boldsymbol{u}\boldsymbol{x}_{2L}(i+1)\right). \end{aligned} \tag{25}$$

Noting that $\boldsymbol{U}^T\boldsymbol{U}$ differs from the identity matrix by only one coefficient in the first position, (25) becomes, after some algebraïc manipulations:

$$\begin{aligned} \boldsymbol{R}_b(i+1) &= \boldsymbol{R}_b(i) - \boldsymbol{x}_{L+1}^T(i+1)\boldsymbol{x}_{L+1}(i+1) \\ &\quad + \boldsymbol{x}_{2L}^T(i+1)\boldsymbol{x}_{2L}(i+1). \end{aligned} \tag{26}$$

Similarly,

$$\begin{aligned} \boldsymbol{R}_f(i+1) &= \boldsymbol{R}_f(i) - \boldsymbol{x}_1^T(i+1)\boldsymbol{x}_1(i+1) \\ &\quad + \boldsymbol{x}_L^T(i+1)\boldsymbol{x}_L(i+1). \end{aligned} \tag{27}$$

Consequently, the covariance matrix for the forward-backward prediction satisfies the recursion

$$\boldsymbol{R}(i+1) = \boldsymbol{R}(i) + \boldsymbol{V}^T(i+1)\boldsymbol{W}(i+1) \tag{28}$$

with

$$\boldsymbol{V}(i) = \begin{bmatrix} -\boldsymbol{x}_1(i) \\ \boldsymbol{x}_L(i) \\ -\boldsymbol{x}_{L+1}(i) \\ \boldsymbol{x}_{2L}(i) \end{bmatrix} \quad \text{and} \quad \boldsymbol{W}(i) = \begin{bmatrix} \boldsymbol{x}_1(i) \\ \boldsymbol{x}_L(i) \\ \boldsymbol{x}_{L+1}(i) \\ \boldsymbol{x}_{2L}(i) \end{bmatrix}. \tag{29}$$

The above recursion for $\boldsymbol{R}(i)$ enables the use of a matrix inversion lemma. The exact update for the inverse becomes

$$\begin{aligned} \boldsymbol{R}^{-1}(i+1) &= \boldsymbol{R}^{-1}(i) - \boldsymbol{R}^{-1}(i)\boldsymbol{V}^T(i+1). \\ &\quad \boldsymbol{Z}^{-1}(i+1)\boldsymbol{W}(i+1)\boldsymbol{R}^{-1}(i) \end{aligned} \tag{30}$$

where

$$\boldsymbol{Z}(i+1) = \boldsymbol{I}_4 + \boldsymbol{V}(i+1)\boldsymbol{R}^{-1}(i)\boldsymbol{W}^T(i+1) \tag{31}$$

is a 4×4 matrix to be inverted at each iteration. Once $\boldsymbol{R}^{-1}(i+1)$ is determined from (30), the vector $\boldsymbol{R}^{-1}(i+1)\boldsymbol{q}(i)$ is computed using a matrix-vector multiplication. The recurrence (30) is typically started by taking $\boldsymbol{R}^{-1}(1) = \delta^{-1}\boldsymbol{I}_p$ as an approximation of the inverse of the first covariance matrix, where δ is a small positive number. One can also use the inverse of the diagonal of $\boldsymbol{R}(1)$ as an estimate of the inverse. These initialization schemes eliminate the matrix inversion at the first block iteration and guarantee good convergence performance. The simulations show faster adaptation when the diagonal approximation is used.

An alternate update of $\boldsymbol{R}(i)$ can be obtained by considering the $2 \times p$ matrices

$$\boldsymbol{V}(i) = \boldsymbol{W}(i) = \begin{bmatrix} \boldsymbol{x}_L(i) \\ \boldsymbol{x}_{2L}(i) \end{bmatrix} \tag{32}$$

in place of (29), that is, we suppose that the shifting property does not eliminate the first row of the matrices $\boldsymbol{X}_f(i)$ and $\boldsymbol{X}_b(i)$. This is equivalent to "keeping" the older information $\boldsymbol{x}_1(i)$ and $\boldsymbol{x}_{L+1}(i)$ and using matrices $\boldsymbol{X}_{fb}(i)$ of size increasing with i. Note however that this does not affect the size of the covariance matrices $\boldsymbol{R}(i)$. The resulting update for $\boldsymbol{R}^{-1}(i+1)$ is similar to (30) with the difference that $\boldsymbol{Z}(i+1)$ is now the 2×2 (symmetric) matrix

$$\boldsymbol{Z}(i+1) = \boldsymbol{I}_2 + \boldsymbol{V}(i+1)\boldsymbol{R}^{-1}(i)\boldsymbol{V}^T(i+1). \tag{33}$$

C. *Implementation via a Gauss-Seidel Iteration*

The matrix inversion for computing the vector $\boldsymbol{R}^{-1}(i)\boldsymbol{q}(i)$ in (22) can be avoided altogether by solving the system

$$\boldsymbol{R}(i)\boldsymbol{z}(i) = \boldsymbol{q}(i) \tag{34}$$

for $\boldsymbol{z}(i)$ via an iterative method (which only requires matrix-vector products), then updating

$$\boldsymbol{a}(i+1) = \boldsymbol{a}(i) + \boldsymbol{z}(i). \tag{35}$$

More precisely, $\boldsymbol{z}(i)$ is replaced by $\boldsymbol{z}^{(k)}(i)$ obtained by applying k iterations

$$\boldsymbol{z}^{(k)}(i) = \boldsymbol{z}^{(k-1)}(i) + \boldsymbol{Q}^{-1}(i)\left(\boldsymbol{q}(i) - \boldsymbol{R}(i)\boldsymbol{z}^{(k-1)}(i)\right) \tag{36}$$

starting with a given vector $\boldsymbol{z}^{(0)}(i)$. Here $\boldsymbol{Q}(i)$ is a matrix approximating $\boldsymbol{R}(i)$. Since the system (34) is symmetric and generally positive definite, and for efficiency reasons, we only consider Gauss-Seidel iterations, i.e.,

$$\boldsymbol{Q}(i) = \boldsymbol{D}(i) + \boldsymbol{L}(i), \tag{37}$$

where $\boldsymbol{D}(i)$ and $\boldsymbol{L}(i)$ are the diagonal and (strictly) lower triangular parts of $\boldsymbol{R}(i)$, respectively. Note that the matrix $\boldsymbol{R}(i)$ is not always diagonally dominant (at least for the input data used), which explains why the Jacobi method (corresponding to $\boldsymbol{Q}(i) = \boldsymbol{D}(i)$) did not converge when applied to (34). In our experiments only 2 or 3 iterations were sufficient to obtain a good approximation of $\boldsymbol{z}(i)$ when starting with $\boldsymbol{z}^{(0)}(i) = 0$. For two iterations, this is equivalent to approximating $\boldsymbol{z}(i)$ by

$$\boldsymbol{z}^{(2)}(i) = (\boldsymbol{D}(i) + \boldsymbol{L}(i))^{-1}. \left(\boldsymbol{q}(i) - \boldsymbol{L}^T(i)(\boldsymbol{D}(i) + \boldsymbol{L}(i))^{-1}\boldsymbol{q}(i)\right). \tag{38}$$

In order to reduce the computational complexity of the algorithm the sum $\boldsymbol{D}(i) + \boldsymbol{L}(i)$ can be directly updated without forming $\boldsymbol{R}(i+1)$, by considering the lower triangular part (including the diagonal) of the matrices in (28), namely

$$\boldsymbol{D}(i+1) + \boldsymbol{L}(i+1) = \boldsymbol{D}(i) + \boldsymbol{L}(i) + \boldsymbol{Y}(i+1) \tag{39}$$

where $\boldsymbol{Y}(i+1)$ is the lower triangular part of $\boldsymbol{V}^T(i+1)\boldsymbol{W}(i+1)$. If $\boldsymbol{V}(i)$ and $\boldsymbol{W}(i)$ are given by (29) then $\boldsymbol{Y}(i+1)$ can be computed cheaply.

D. *Computational Complexity*

The computational complexity of the proposed algorithms compares favorably with several of the recently proposed algorithms. The BMCAI has two different implementation schemes. The first one is based on the matrix inversion lemma and the second one uses the Gauss-Seidel method. The results in Table 1 show that the Gauss-Seidel scheme outperforms the algorithms based on updates of the inverse of the covariance matrix.

Algorithm	Multiplies	Additions
BMCA	$4Lp + p$	$4Lp$
BMCA-2	$8Lp + 2N$	$8Lp + p - 2L - 2$
BMCA-3	$8Lp + 4p + 6L + 9$	$8L(p-1) + 8L + 5p - 2$
BMCAI-1	$4Lp + 6p^2 + 10p + 2$	$4Lp + 6p^2 + 2p$
BMCAI-2	$p(4N - 3p/2 + 3/2)$	$p(4N - 3p/2 - 1/2) + 1$

Table 1: Computational Complexity of BMCA algorithms. BMCAI-1 utilizes the matrix inversion lemma and BMCAI-2 uses the Gauss-Seidel iteration. $L = N - p$.

IV. Simulation Results

A. *Spectral estimates*

The performance of the proposed algorithms is examined in (AR) spectral estimation using some "benchmark" examples. The BMCAI was used to examine the ability to resolve closely-spaced peaks. The PSD obtained using the regular fixed μ BMCA is compared with the PSD obtained from the BMCAI. The results show that the optimum algorithms give improved performance relative to resolution, bias, spectral variance and line splitting without being affected from the additive noise (robustness). The processes employed were closely spaced peaks from multiple sinusoids in white noise or non-sinusoidal sequences in additive exponential decays.

B. *Two Closely-Spaced Cosines in additive exponential decay.*

Nikias and Scott [12] examined the ability of the energy weighted covariance method to resolve sinusoids corrupted by additive exponentials. Such type of processes appear in the field of geophysics and biomedical applications. Most of these processes are modeled by sinusoids corrupted by additive exponentials, or by sinusoids of fixed frequency whose amplitude is modulated by an envelope function. Therefore, the ability of the BMCAI with individual adaptation of parameters to tolerate additive exponential decay-type non-stationarity in sinusoidal data was tested. By using a sequence consisting of two closely-spaced cosine waves in additive exponential decay of the form $x(n) = cos(2\pi(n-1)0.25 + 3\pi 0.25) + cos(2\pi(n-1)0.234375) + q^{(n-1)}$ where $q = 0.1, 0.5, 0.9$ and we performed a series of experiments using a short record of samples for the three different exponentials (Figure 1). The AR model order employed was $p = 5$. The results revealed that the BMCAI is robust in the sense that its ability to resolve closely spaced sinusoids was

not affected by the additive exponential. Note also that the resolution is very high and the frequency bias is small.

C. *Ten Closely-Spaced Spectral Peaks of Sinusoidal Process*

In order to verify the performance of the BMCAI based on Gauss-Seidel iterations, an input time series was generated. In particular, we investigate the resolution of closely-spaced spectral peaks of a process consisting of ten sinusoids in additive white noise. The process is defined as follows:

$$x(n) = \sum_{i=1}^{10} A_i \cos(\omega_i n) + W(n) \qquad (40)$$

for $n = 1, ..., 32$, with $A_i = 0.1i$, $\omega_i = \frac{2\pi(10+1.5(i-1))}{f_s}$ and $Q = 10^{-4}$ (noise variance). Here $f_s = 64$ is the sampling frequency (in Hertz) and $W(n)$ a pseudo-random white-noise sequence. The same time series was used to demonstrate the superiority of the combined forward-backward prediction method as compared to the unidirectional method (that is the method that estimates the AR parameters using forward (backward) prediction samples only). The predictors order was taken equal to 32. The plot in Figure 2 are formed by overlapping the spectra obtained using the BMCAI with individual adaptation of parameters based on Gauss-Seidel iterations, for 10 independent realizations. Each realization is a 100-sample record of the above input time series. The relative phases change randomly from realization to realization. The algorithm clearly identifies the unknown sinusoids, although they are close enough to one another and the available data records have relatively short length.

D. *Absence of Line Splitting*

Marple [13] has reported that the LS generated spectra does not suffer from line splitting. According to Fougere *et al.* [14], the worst line splitting effects occur in Burg's algorithm [5] for long portions of sinusoidal data an odd number of quarter cycles with initial phase 45^o. In order to show that the AR spectra formed using the BMCAI do not suffer from line splitting we have reconstructed an example provided by Fougere in which a unit amplitude sine wave in additive white noise ($f = 26.25$, $f_s = 100$, $\phi = 45^o$, $N = 101$, $p = 25$, $Q = 10^{-4}$ (noise variance)) was employed. The total number of samples used for the BMCA was 101. Figure 3 shows that the BMCAI generated spectra do not suffer from line splitting and locate the sinusoid at the correct frequency. Running a considerable amount of simulations, line splitting tendency of the new BMCAI has not been observed.

E. *Two Closely-Spaced Spectral Peaks from an AR(4) Process*

This simulation is concerned with the performance of the BMCAI with individual adaptation of parameters using a harmonic fourth-order AR process with parameters $a_1 = -2.7607$, $a_2 = 3.8106$, $a_3 = 2.6535$, $a_4 = 0.9238$ driven by white Gaussian process. Figure 4(a) shows an ensemble of 20 PSD's estimated from 20 independent runs of the BMCAI. Figure 4(b) shows the average PSD obtained from the 20 realizations of optimum BMCAI. The total number of samples used for each run was 40. The white Gaussian has variance $Q = 1.0$. The model order was set at the correct value $p = 4$. Note that the original and average estimated PSD are very close.

V. Conclusions

In this paper, the formulation of the block modified covariance algorithms which incorporates optimum convergence factors, one for all parameters, two for all parameters but different for forward and backward prediction and the most general case where each parameter has an *individual* convergence factor have been presented. The convergence factors are optimally selected to minimize the combined forward-backward squared error in each block. Two algorithms for estimating the factors were proposed and investigated. One of them uses two time-varying convergence factors for the forward and backward linear prediction parts; the other one is the individual adaptation algorithm and uses one factor for each parameter.

Two different methods for implementing the BMCAI have been investigated in this paper, that eliminate the matrix inversion and reduce considerably the computational complexity. The proposed algorithms have been applied in AR spectral estimation and the simulations support the conclusion that the estimates have high resolution, low variance and do not suffer from line splitting or frequency bias.

References

[1] A. S. Spanias, "Block time and frequency domain modified covariance algorithms for spectral analysis," *IEEE Transactions on Signal Processing*, vol. ASSP-41, no. 6, pp. 2131–2140, June 1993.

[2] S. L. Marple, "A new autoregressive spectrum analysis algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, no. 6, pp. 441–454, Aug. 1980.

[3] T. J. Ulrych and R. W. Clayton, "Time series modelling and maximum entropy," *Phys. Earth Plan. Int.*, vol. , no. 12, pp. 188–200, Aug. 1976.

[4] A. H. Nuttal, "Spectral analysis of a univariate process with bad data points via maximum

entropy and linear predictive techniques," *Naval Underwater Systems Center TR-5303, New London*, vol. , no. , p. , Mar. 76.

[5] J. P. Burg, "Maximum entropy spectral analysis," *Proc. 37th Meet. of the Soc. Expl. Geoph.*, vol. , no. , p. , 1967.

[6] S. L. Marple, "A fast computational algorithm for the modified covariance method of linear prediction," *Digital Signal Processing 1*, vol. New York: Academic, no. 6, pp. 124–133, 1991.

[7] C. Demeure and L. Scharf, "Lattice algorithms for computing qr and cholesky factors in the least squares theory of linear prediction," *Univ. Colorado, Tech. Rep.*, vol. , no. , p. , Sep 1987.

[8] N. Kalouptsidis and S. Theodoridis, "Fast adaptive least squares algorithms for power spectral astimation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, no. , pp. 661–667, May 1987.

[9] K. Berberidis and S. Theodoridis, "Efficient symmetric algorithms for the modified covariance method for autoregressive spectral analysis," *IEEE Transactions on Signal Processing*, vol. ASSP-41, no. 7, pp. 43–54, Jul. 1986.

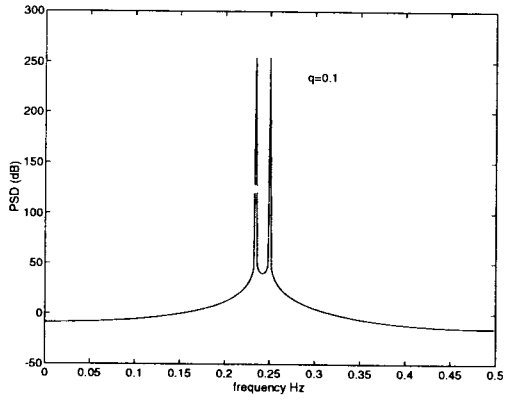
[10] W. Mikhael and F. Wu, "A fast block fir adaptive digital filtering algorithm with individual adaptation of parameters," *IEEE Transactions on Circuits and Systems*, vol. cas-36, no. 1, pp. 1–10, Jan. 1989.

[11] T. Wang and C. L. Wang, "On the optimum design of the block adaptive fir digital filter," *IEEE Transactions on Signal Processing*, vol. ASSP-41, no. 6, pp. 2131–2140, Oct. 1987.

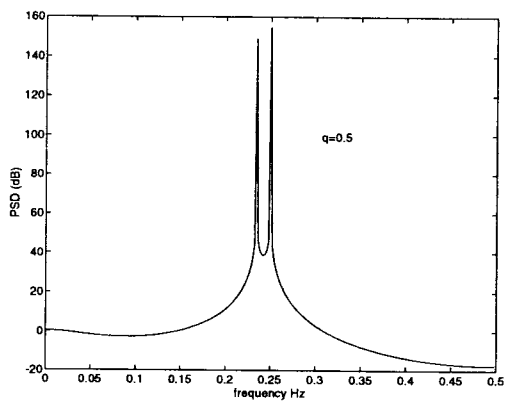
[12] C. Nikias and P. Scott, "The covariance least-squares algorithm for spectal estimation of processes of short data length," *IEEE Trans. on Geoscince and Remote Sensing*, vol. GE-21, no. 2, pp. 180–191, 1983.

[13] S. L. Marple, *Digital Spectral Analysis with Applications*. Englewood Cliffs, N.J.: Prentice Hall, 1987.

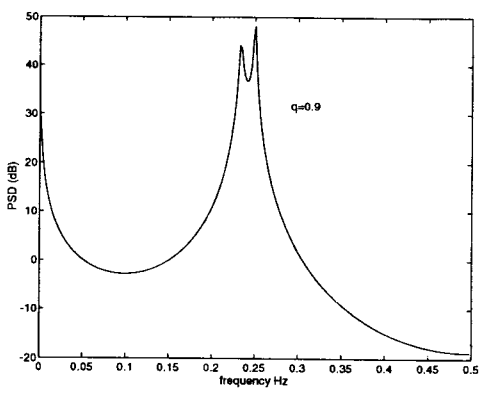
[14] P. Fougere, "Spontaneous line splitting in maximum entropy power spectrum analysis," *Phys. Earth Plan. Inter.*,, vol. 12, no. 2, pp. 201–207, Aug. 1976.



(a)

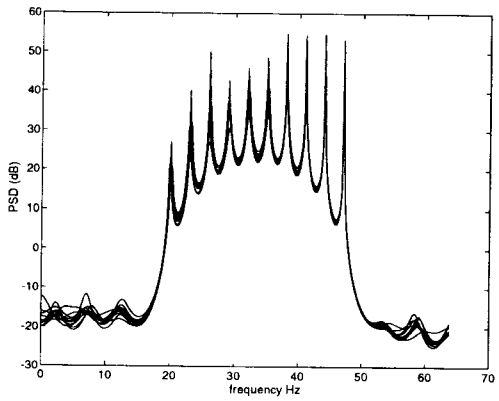


(b)

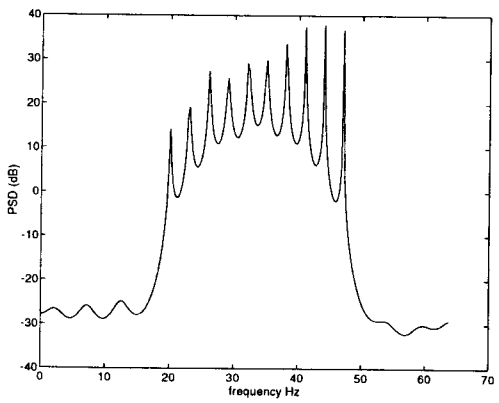


(c)

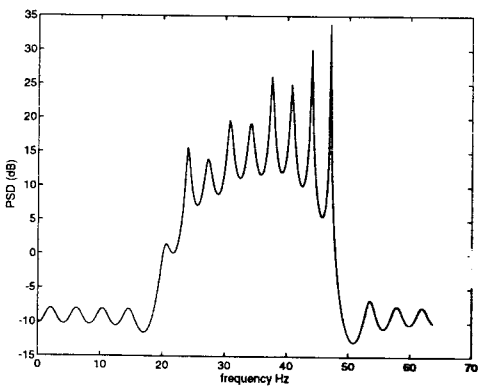
Figure 1: PSD of two closely spaced sinusoids in additive exponential transients computed using the BMCAI based on Gauss-Seidel iterations. The parameters are $N = 20$, $p = 5$, $r = 5$, $S = 1$, with $q = 0.1$ (a), 0.5 (b) and 0.9 (c). The total number of iterations was 20 and the total number of samples used was 40.



(a)



(b)



(c)

Figure 2: PSD estimation using the BMCAI based on Gauss-Seidel iterations with 10 realizations of 100-sample records, predictors order equal to 32 and SNR=42dB (a) and its average (b), and using the fixed $\mu = 0.001$ (c).

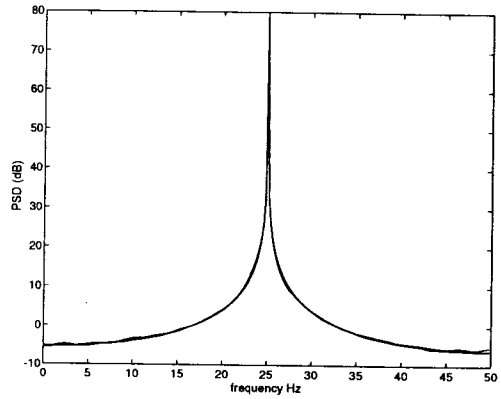
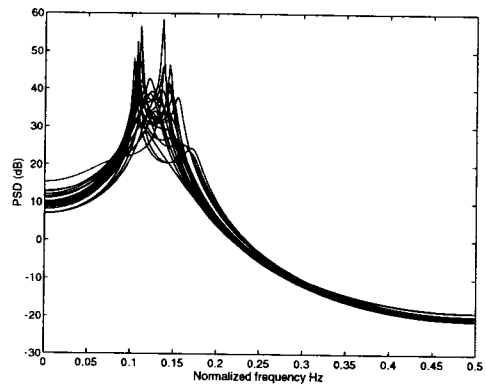
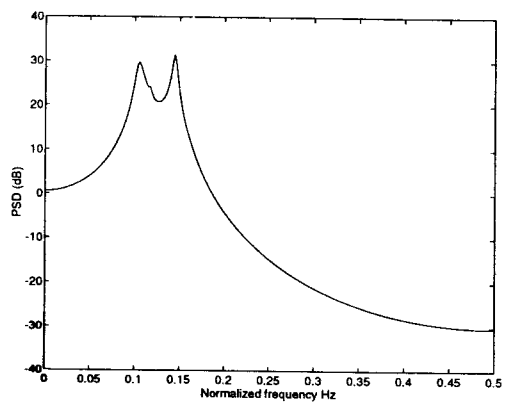


Figure 3: PSD of a sinusoid plus white noise AR(25) with parameters used $N = 50$, $p = 25$, $S = 1$, $r = 1$ and 101 samples. Note the absence of line splitting.



(a)



(b)

Figure 4: **PSD of Two Closely-Spaced Spectral Peaks of Non-Sinusoidal Process using an AR estimate obtained from the BMCAI algorithm with individual adaptation of parameters based on Gauss-Seidel iterations. Ensemble of 20 realizations of 40 sample records (a) and its average (b).**