

ON RECURSIVE AND FAST RECURSIVE COMPUTATION OF THE CAPON SPECTRUM

Jacob Benesty¹, Jingdong Chen², and Yiteng (Arden) Huang²

¹: Université du Québec, INRS-EMT
800 de la Gauchetière Ouest, Suite 6900
Montréal, Québec, H5A 1K6, Canada
e-mail: benesty@emt.inrs.ca

²: Bell Labs, Lucent Technologies
600 Mountain Avenue
Murray Hill, NJ, 07974, USA
e-mail: {jingdong,arden}@research.bell-labs.com

ABSTRACT

The Capon spectrum, which is known to have better resolution than the periodogram, has been widely used in various applications. Normally, the Capon spectrum is estimated through the direct computation of the inverse of the data correlation (or covariance) matrix. This so-called direct inverse approach is, however, computationally very expensive due to the high computational cost involved in the matrix inversion. This paper deals with fast and efficient algorithms in computing the Capon spectrum. Inspired from the recursive idea established in the area of adaptive signal processing, we first derive a recursive Capon algorithm. This new algorithm does not require an explicit matrix inversion, and hence is more efficient to implement than the direct inverse method. We then develop a fast version of the recursive algorithm, which can further reduce the complexity of the recursive one by an order of magnitude.

Index Terms— Capon, MVDR, Spectral Estimation, Recursive Least Squares, Linear Prediction.

1. INTRODUCTION

Spectral estimation, which endeavors to determine the spectral content of a signal from a finite set of measurements, plays a critical role in various applications. This problem has attracted a considerable amount of research attention over the last century [1], [2]. Among the numerous techniques that were developed, the Capon method can be considered as one of the most fundamental approaches, which is known to have better resolution than the periodogram and thus has been widely adopted for many applications.

Briefly, the Capon method for spectral estimation is based on a filterbank decomposition: the spectrum of a signal is estimated in each band by a simple filter design subject to some constraints [3], [4]. Let $x(n)$ be a zero-mean random process with a power spectral density (PSD) $S_{xx}(\omega)$, where ω ($0 \leq \omega < 2\pi$) is the angular frequency. Suppose that we want to design a finite impulse response (FIR) filter that passes the frequency ω_k in $x(n)$ without distortion, and meanwhile, attenuates all the other frequencies as much as possible. This will lead to the well-known Capon filter, which is mathematically written as

$$\mathbf{h}_k^{\text{Capon}} = \arg \min_{\mathbf{h}_k} E[|y_k(n)|^2] = \arg \min_{\mathbf{h}_k} \mathbf{h}_k^H \mathbf{R}_{xx} \mathbf{h}_k \quad (1)$$

$$\text{subject to } \mathbf{h}_k^H \mathbf{f}_k = 1,$$

where $y_k(n)$, $k = 0, 1, \dots, K-1$, is the output of the FIR filter, $\mathbf{R}_{xx} = E[\mathbf{x}(n)\mathbf{x}^H(n)]$ is the covariance matrix of $x(n)$,

$$\begin{aligned} \mathbf{h}_k &= [h_{k,0} \quad h_{k,1} \quad \dots \quad h_{k,L-1}]^T, \\ \mathbf{x}(n) &= [x(n) \quad x(n-1) \quad \dots \quad x(n-L+1)]^T, \end{aligned}$$

$$\mathbf{f}_k = [1 \quad \exp(j\omega_k) \quad \dots \quad \exp[j(L-1)\omega_k]]^T,$$

T and H denote, respectively, the transpose and conjugate transpose, and $\omega_k = 2\pi k/K$.

The solution to above optimization problem can be deduced as (assuming that \mathbf{R}_{xx}^{-1} exists) [3], [4]:

$$\mathbf{h}_k^{\text{Capon}} = \frac{\mathbf{R}_{xx}^{-1} \mathbf{f}_k}{\mathbf{f}_k^H \mathbf{R}_{xx}^{-1} \mathbf{f}_k}. \quad (2)$$

The power of $x(n)$ in the passband of the Capon filter centered on ω_k is easily written as:

$$E[|y_k(n)|^2] = \frac{1}{\mathbf{f}_k^H \mathbf{R}_{xx}^{-1} \mathbf{f}_k}. \quad (3)$$

The spectrum $S_{xx}(\omega)$ at frequency ω_k can then be determined as:

$$S_{xx}(\omega_k) = \frac{c}{\mathbf{f}_k^H \mathbf{R}_{xx}^{-1} \mathbf{f}_k}, \quad (4)$$

where the factor c is added for properly scaling the Capon power estimator to obtain the spectral density. The scaling factor is typically determined based on the filter bandwidth. It can be seen from (2) that the Capon filter is data dependent, so the scaling factor may not necessarily be data and frequency independent. Many methods have been developed for determining the scaling factor c , and the most accurate one is the method provided in [5]. However, since our focus in this paper is on fast computation of Capon algorithm rather than the Capon spectral estimator itself, we take the simplest method given in [2] and set $c = K$, where K is the number of bandpass filters. For a good detailed discussion on this issue, the reader is invited to consult [5].

In practice, the covariance matrix \mathbf{R}_{xx} has to be estimated. Suppose that we replace the expectation operation by the exponentially weighted sample average. We then have

$$\hat{\mathbf{R}}_{xx}(n) = \sum_{m=0}^n \lambda^{n-m} \mathbf{x}(m)\mathbf{x}^H(m) \quad (5)$$

where λ ($0 < \lambda < 1$) is a forgetting factor. An estimate of the Capon filter and spectrum at frequency ω_k and time n can then be written as

$$\hat{\mathbf{h}}_k^{\text{Capon}}(n) = \frac{\hat{\mathbf{R}}_{xx}^{-1}(n) \mathbf{f}_k}{\mathbf{f}_k^H \hat{\mathbf{R}}_{xx}^{-1}(n) \mathbf{f}_k}, \quad (6)$$

and

$$\hat{S}_{xx}(\omega_k, n) = \frac{(1-\lambda)K}{\mathbf{f}_k^H \hat{\mathbf{R}}_{xx}^{-1}(n) \mathbf{f}_k}. \quad (7)$$

We deduce from (6) and (7) that:

$$K(1 - \lambda)\hat{\mathbf{R}}_{xx}(n)\hat{\mathbf{h}}_k^{\text{Capon}}(n) = \hat{S}_{xx}(\omega_k, n)\mathbf{f}_k. \quad (8)$$

Taking into account all frequencies ω_k , $k = 0, 1, \dots, K - 1$, we can rewrite (8) into the following form:

$$K(1 - \lambda)\hat{\mathbf{R}}_{xx}(n)\hat{\mathbf{H}}(n) = \mathbf{F}\hat{\mathbf{S}}_{xx}(\omega_k, n), \quad (9)$$

where

$$\begin{aligned} \hat{\mathbf{H}}(n) &= \begin{bmatrix} \hat{\mathbf{h}}_0^{\text{Capon}}(n) & \cdots & \hat{\mathbf{h}}_{K-1}^{\text{Capon}}(n) \end{bmatrix}, \\ \mathbf{F} &= \begin{bmatrix} \mathbf{f}_0 & \mathbf{f}_1 & \cdots & \mathbf{f}_{K-1} \end{bmatrix}, \\ \hat{\mathbf{S}}_{xx}(\omega_k, n) &= \text{diag} \left\{ \hat{S}_{xx}(\omega_0, n), \dots, \hat{S}_{xx}(\omega_{K-1}, n) \right\}. \end{aligned}$$

For $K = L$, \mathbf{F} is the Fourier matrix and $\mathbf{F}^H \mathbf{F} = K\mathbf{I}$ so $\mathbf{F}^{-1} = \frac{1}{K}\mathbf{F}^H$. We then obtain the following interesting decomposition:

$$\hat{\mathbf{S}}_{xx}(\omega_k, n) = (1 - \lambda)\mathbf{F}^H \hat{\mathbf{R}}_{xx}(n) \hat{\mathbf{H}}(n). \quad (10)$$

2. A RECURSIVE COMPUTATION OF THE INVERSE SPECTRUM

The estimation of the Capon spectrum using (7) and (10) requires the computation of the inverse of the covariance matrix, which can be computationally very expensive. The aim of this section is to develop a recursion for the Capon algorithm so that the spectrum can be estimated more efficiently.

The covariance matrix of the signal $x(n)$ can be computed recursively,

$$\hat{\mathbf{R}}_{xx}(n) = \lambda\hat{\mathbf{R}}_{xx}(n-1) + \mathbf{x}(n)\mathbf{x}^H(n). \quad (11)$$

By using the matrix inversion lemma [6], $\hat{\mathbf{R}}_{xx}^{-1}(n)$ can also be computed recursively:

$$\hat{\mathbf{R}}_{xx}^{-1}(n) = \lambda^{-1}\hat{\mathbf{R}}_{xx}^{-1}(n-1) - \lambda^{-2}\varphi(n)\mathbf{g}'(n)\mathbf{g}'^H(n), \quad (12)$$

where

$$\mathbf{g}'(n) = \hat{\mathbf{R}}_{xx}^{-1}(n-1)\mathbf{x}(n) \quad (13)$$

is the *a priori* Kalman gain vector and

$$\varphi(n) = \frac{\lambda}{\lambda + \mathbf{x}^H(n)\hat{\mathbf{R}}_{xx}^{-1}(n-1)\mathbf{x}(n)}. \quad (14)$$

The *a posteriori* Kalman gain vector $\mathbf{g}(n) = \hat{\mathbf{R}}_{xx}^{-1}(n)\mathbf{x}(n)$ is related to $\mathbf{g}'(n)$ by [7]:

$$\mathbf{g}(n) = \lambda^{-1}\varphi(n)\mathbf{g}'(n). \quad (15)$$

Now, if we pre- and post-multiply both sides of (12) by \mathbf{f}_k^H and \mathbf{f}_k respectively and with the help of (7), we get:

$$\frac{(1 - \lambda)K}{\hat{S}_{xx}(\omega_k, n)} = \frac{\lambda^{-1}(1 - \lambda)K}{\hat{S}_{xx}(\omega_k, n-1)} - \lambda^{-2}\varphi(n) \left| \mathbf{f}_k^H \mathbf{g}'(n) \right|^2, \quad (16)$$

hence,

$$\hat{S}_{xx}^{-1}(\omega_k, n) = \lambda^{-1}\hat{S}_{xx}^{-1}(\omega_k, n-1) - \lambda^{-2}\varphi'(n) |v_{g,k}(n)|^2, \quad (17)$$

where

$$\varphi'(n) = (1 - \lambda)^{-1}K^{-1}\varphi(n) \quad (18)$$

and

$$v_{g,k}(n) = \mathbf{f}_k^H \mathbf{g}'(n). \quad (19)$$

Expression (17) shows how the inverse spectrum of the signal $x(n)$ at frequency ω_k and time n can be computed recursively, from its value at time $n - 1$, when a new data sample is available. The recursive algorithm can be summarized as follows:

- At the beginning, initialize $\hat{\mathbf{R}}_{xx}^{-1}(0) = E_0^{-1}\mathbf{I}$ (E_0 is the signal energy), and $\hat{S}_{xx}^{-1}(\omega_k, 0) = L/[E_0(1 - \lambda)K]$.
- When a new data sample is measured at time n , update $\mathbf{g}'(n)$, $\varphi(n)$, $\hat{\mathbf{R}}_{xx}^{-1}(n)$, and $\varphi'(n)$ according to (13), (14), (12), and (18), respectively.
- At time n , estimate the inverse spectrum according to (17).

From (17), we see that the estimation of the inverse Capon spectrum requires the computation of one inner product. The corresponding complexity is proportional to KL . For $K = L$, this complexity is proportional to L^2 , which is quite high for practical applications. A natural question then arises: can we further reduce the number of operations to make it linear with respect to L ? In the next section, we will discuss an efficient recursive Capon algorithm.

3. A FAST RECURSIVE ALGORITHM

In this section, we are going to show that the inner product $v_{g,k}(n) = \mathbf{f}_k^H \mathbf{g}'(n)$ can be computed recursively with a couple of multiplications only at each iteration, instead of L . As a result, the complexity of the entire algorithm will be reduced significantly since $v_{g,k}(n)$ has to be evaluated K times for every time sample n .

3.1. A Fast Algorithm Based on Linear Prediction

It is well-known that the *a priori* Kalman gain vector of order $L + 1$ can be computed in two different ways [7]:

$$\mathbf{g}'_1(n) = \begin{bmatrix} \mathbf{g}'(n) \\ 0 \end{bmatrix} + \frac{e_b(n)}{E_b(n-1)} \begin{bmatrix} -\mathbf{b}(n-1) \\ 1 \end{bmatrix} \quad (20)$$

$$= \begin{bmatrix} 0 \\ \mathbf{g}'(n-1) \end{bmatrix} + \frac{e_a(n)}{E_a(n-1)} \begin{bmatrix} 1 \\ -\mathbf{a}(n-1) \end{bmatrix}, \quad (21)$$

where $\mathbf{a}(n)$ and $\mathbf{b}(n)$ are, respectively, the forward and backward predictors of order L , $e_a(n)$ and $e_b(n)$ are the *a priori* forward and backward prediction error signals, and $E_a(n)$ and $E_b(n)$ are the forward and backward prediction error energies.

Consider the following vector of length $L + 1$:

$$\begin{aligned} \mathbf{f}_{1,k} &= \begin{bmatrix} 1 & \exp(j\omega_k) & \cdots & \exp(j\omega_k L) \end{bmatrix}^T \\ &= \begin{bmatrix} \mathbf{f}_k^T & \exp(j\omega_k L) \end{bmatrix}^T = \begin{bmatrix} 1 & \exp(j\omega_k)\mathbf{f}_k^T \end{bmatrix}^T. \end{aligned}$$

If we pre-multiply both sides of (20) and (21) by $\mathbf{f}_{1,k}^H$, we obtain the recursion:

$$\begin{aligned} v_{g,k}(n) &= \exp(-j\omega_k)v_{g,k}(n-1) - \\ &\frac{e_b(n)}{E_b(n-1)} [\exp(-j\omega_k L) - v_{b,k}(n-1)] + \\ &\frac{e_a(n)}{E_a(n-1)} [1 - \exp(-j\omega_k)v_{a,k}(n-1)], \end{aligned} \quad (22)$$

where

$$v_{a,k}(n-1) = \mathbf{f}_k^H \mathbf{a}(n-1), \quad (23)$$

$$v_{b,k}(n-1) = \mathbf{f}_k^H \mathbf{b}(n-1). \quad (24)$$

In order for (22) to be efficient, (23) and (24) need to be computed recursively. This can be easily done thanks to the update equations of the forward and backward predictors:

$$\mathbf{a}(n) = \mathbf{a}(n-1) + \frac{e_a^*(n)}{\alpha(n-1)} \mathbf{g}'(n-1), \quad (25)$$

$$\mathbf{b}(n) = \mathbf{b}(n-1) + \frac{e_b^*(n)}{\alpha(n)} \mathbf{g}'(n), \quad (26)$$

where the superscript $*$ is the complex conjugate operator and $\alpha(n) = \lambda/\varphi(n)$. Now, if we pre-multiply both sides of (25) and (26) by \mathbf{f}_k^H , we deduce the two recursions:

$$v_{a,k}(n) = v_{a,k}(n-1) + \frac{e_a^*(n)}{\alpha(n-1)} v_{g,k}(n-1), \quad (27)$$

$$v_{b,k}(n) = v_{b,k}(n-1) + \frac{e_b^*(n)}{\alpha(n)} v_{g,k}(n). \quad (28)$$

The two previous expressions should be used in (22). Therefore, the inner product $v_{g,k}(n)$ can be estimated with roughly six complex multiplications, instead of L . To summarize, the fast recursive algorithm estimates the inverse spectrum according to the following iterative steps.

- At the beginning, initialize $\mathbf{g}'(0) = \mathbf{a}(0) = \mathbf{b}(0) = \mathbf{0}$, $\alpha(0) = \lambda$, $E_a(0) = E_0$, $E_b(0) = E_0\lambda^{-L}$, $v_{a,k}(0) = v_{b,k}(0) = v_{g,k}(0) = 0$, and $S_{xx}^{-1}(\omega_k, 0) = L/[E_0(1 - \lambda)K]$, $\forall k$.
- At time n , compute

$$\begin{aligned} e_a(n) &= x(n) - \mathbf{a}^H(n-1)\mathbf{x}(n-1), \\ \alpha_1(n) &= \alpha(n-1) + |e_a(n)|^2/E_a(n-1), \\ \begin{bmatrix} \mathbf{t}(n) \\ m(n) \end{bmatrix} &= \begin{bmatrix} 0 \\ \mathbf{g}'(n-1) \end{bmatrix} + \begin{bmatrix} 1 \\ -\mathbf{a}(n-1) \end{bmatrix} \cdot \\ &\quad e_a(n)/E_a(n-1), \\ E_a(n) &= \lambda [E_a(n-1) + |e_a(n)|^2/\alpha(n-1)], \\ \mathbf{a}(n) &= \mathbf{a}(n-1) + \mathbf{g}'(n-1)e_a^*(n)/\alpha(n-1), \\ e_b(n) &= x(n-L) - \mathbf{b}^H(n-1)\mathbf{x}(n), \\ \mathbf{g}'(n) &= \mathbf{t}(n) + \mathbf{b}(n-1)m(n), \\ \alpha(n) &= \alpha_1(n) - e_b^*(n)m(n), \\ E_b(n) &= \lambda [E_b(n-1) + |e_b(n)|^2/\alpha(n)], \\ \mathbf{b}(n) &= \mathbf{b}(n-1) + \mathbf{g}'(n)e_b^*(n)/\alpha(n), \\ \varphi'(n) &= \frac{\varphi(n)}{(1-\lambda)K}. \end{aligned}$$

- At time n , estimate $v_{a,k}(n-1)$, $v_{b,k}(n-1)$, $v_{g,k}(n)$, and $\hat{S}_{xx}^{-1}(\omega_k, n)$ according to (27), (28), (22), and (17), respectively.

3.2. Complexity Analysis

Now let us compare the computational complexity of the direct-inverse, the recursive, and the fast recursive Capon algorithms. Here the computational complexity is evaluated in terms of the number of real-valued multiplications/divisions required for the implementation of each algorithm. The number of additions/subtractions are neglected because they are much quicker to compute in most generic hardware platforms. We assume that complex-valued multiplications are transformed into real-valued multiplications. The multiplication between a real and complex numbers requires 2 real-valued

multiplications. The multiplication between two complex numbers needs 4 real-valued multiplications. The division between a complex number and a real number requires 2 real-valued multiplications.

Suppose that the observation signal is real-valued and a spectral estimate has to be made every N samples. The direct-inverse approach achieves the spectral estimate in two steps. It first computes the signal correlation matrix according to (5). This step requires $N(L^2 + L)$ multiplications. It then estimates the Capon spectrum using (7). If we assume that the inverse of the correlation matrix is computed through the LU decomposition, which requires $L^3 - L$ multiplications, it is trivial to deduce the grand total, for estimating the inverse spectrum by the direct-inverse method, of $L^3 + (N + 2K)L^2 + (N + 4K - 1)L + K$ multiplications.

For the recursive algorithm, the inverse spectrum depends on the *a priori* Kalman gain vector $\mathbf{g}'(n)$, which can be computed at each iteration, using linear prediction techniques. This step involves $16L + 15$ multiplications. Equation (17), which requires the calculation of one inner product, involves $(2L + 4)K$ multiplications for estimating the inverse spectra for all frequencies ω_k , $k = 0, 1, \dots, K-1$. The total cost for N samples is, therefore, $N(2LK + 16L + 4K + 15)$ multiplications.

The fast recursive algorithm also requires the computation of the *a priori* Kalman gain vector $\mathbf{g}'(n)$, which involves $16L + 15$ multiplications. But this technique can estimate the inverse spectrum with only $20K$ multiplications after knowing the Kalman gain vector. The total complexity for N samples is, therefore, $N(16L + 15 + 20K)$.

If we assume $N = L = K$, the computational complexities for the direct-inverse, recursive, and fast recursive Capon algorithms are, respectively, $4L^3 + 5L^2$, $2L^3 + 20L^2 + 15L$, and $36L^2 + 15L$ multiplications. Since L is often in the order of a few tens or hundreds of taps, we clearly see that the recursive algorithm is computationally less expensive than the direct-inverse approach and the fast recursive algorithm is much more efficient than both the recursive and direct inverse approaches.

3.3. Discussion on the Bias Removal

Having discussed the complexity issue, we now examine the error-propagation effect, another important problem that needs much attention when any fast algorithm is developed.

From Section 3.1, we can see that the proposed fast algorithm requires two initializations. One is for the inverse spectrum $[\hat{S}_{xx}^{-1}(\omega_k, 0)]$. The other is for the prediction error energies $[E_a(0)$ and $E_b(0)]$, which are involved in the efficient computation of the Kalman gain vector, which in turn is used to estimate the inverse spectrum. Both initializations depend on the energy (E_0) of the signal $x(n)$. Because of this interlink process and the fact that the two initializations are not perfectly synchronized, one interesting phenomenon appears during the update: a bias is introduced in the inverse spectrum estimation, which grows with the time index n . As an example, we illustrate this phenomenon with a zero-mean white Gaussian signal $x(n)$ with a variance of $\sigma_x^2 = 1$, $L = K = 100$, $\lambda = 1 - 1/(5L)$, and for $n = 1000$. Figure 1 shows the estimates of the inverse spectrum with the recursive [Fig. 1(a)] and fast recursive [Fig. 1(b)] algorithms. We can notice that the two inverse spectra are identical but the y -axis scale is different. This difference is due to the bias. Therefore, inverting the estimate obtained with the fast recursive algorithm will give a wrong result for the spectrum estimation of the signal $x(n)$.

From the recursive expression of the inverse spectrum given in (17), we have

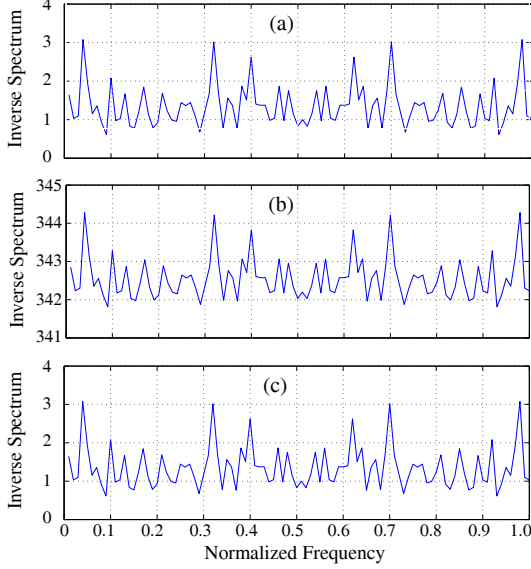


Fig. 1. Inverse spectrum of a white Gaussian signal with $\sigma_x^2 = 1$, $L = K = 100$, $\lambda = 1 - 1/(5L)$, and $n = 1000$: (a) Recursive algorithm; (b) Fast recursive algorithm; and (c) Fast recursive algorithm after bias removal.

$$\begin{aligned}\hat{S}_{xx}^{-1}(\omega_k, n) &= \lambda^{-1} \hat{S}_{xx}^{-1}(\omega_k, n-1) - \lambda^{-2} \xi(n) \\ &= \lambda^{-2} \hat{S}_{xx}^{-1}(\omega_k, n-2) - \lambda^{-3} \xi(n-1) - \lambda^{-2} \xi(n)\end{aligned}\quad (29)$$

where $\xi(n) = \varphi'(n) |v_{g,k}(n)|^2$. Continuing this recursion, we deduce that

$$\hat{S}_{xx}^{-1}(\omega_k, n) = \frac{L\lambda^{-n}}{(1-\lambda)KE_0} - \lambda^{-2} \sum_{m=0}^{n-1} \lambda^{-m} \xi(n-m). \quad (30)$$

The first term of the right hand side of the previous equation depends on the initialization E_0^{-1} while the second term depends on E_0 when computed with the fast algorithm (but it does not appear explicitly). Now suppose that we initialize E_0 with E_1 , which satisfies $E_1 = E_0 + \delta$ (where δ is a small positive number and $\delta \ll E_1$). Replacing E_0 in (30) with E_1 and using the approximation $\frac{1}{1-\delta/E_1} \approx 1 + \delta/E_1$, we get the corresponding spectrum estimate,

$$\begin{aligned}\hat{S}_{xx}^{-1}(\omega_k, n) &= \frac{L\lambda^{-n}}{(1-\lambda)KE_1} + \frac{\delta L\lambda^{-n}}{(1-\lambda)KE_1^2} \\ &\quad - \lambda^{-2} \sum_{m=0}^{n-1} \lambda^{-m} \varphi'(n-m) |v_{g,k}(n-m)|^2 \\ &= \frac{\delta L\lambda^{-n}}{(1-\lambda)KE_1^2} + \hat{S}_{xx}^{-1}(\omega_k, n) = \Delta_b(n) + \hat{S}_{xx}^{-1}(\omega_k, n).\end{aligned}\quad (31)$$

The term $\Delta_b(n)$ represents the bias which grows exponentially with n . Expression (31) shows very clearly that even a very small mismatch between E_0 and E_1 (or equivalently a very small δ) will introduce an avoidable bias. Obviously, a technique is needed to remove it.

According to (31), we have:

$$\frac{1}{K} \sum_{k=0}^{K-1} \hat{S}_{xx}(\omega_k, n) = \frac{1}{K} \sum_{k=0}^{K-1} \left[\hat{S}_{xx}^{-1}(\omega_k, n) - \Delta_b(n) \right]^{-1}$$

$$= (1-\lambda) \sum_{m=0}^n \lambda^{n-m} |x(m)|^2 = \sigma_x^2(n),$$

and $\Delta_b(n) < \min_k \hat{S}_{xx}^{-1}(\omega_k, n)$. Taking into account the information from the two previous expressions, the following simple iterative algorithm

$$e(n, i) = \frac{K}{L} - \frac{1}{K\sigma_x^2(n)} \cdot \sum_{k=0}^{K-1} \left[\hat{S}_{xx}^{-1}(\omega_k, n) - \Delta_b(n, i-1) \right]^{-1}, \quad (32)$$

$$\Delta_b(n, i) = \Delta_b(n, i-1) + \frac{0.1}{\sigma_x^2(n)} e(n, i), \quad (33)$$

with $\Delta_b(n, 0) = \min_k \hat{S}_{xx}^{-1}(\omega_k, n) - 1/\sigma_x^2(n)$, finds the bias with less than 100 iterations. Let's take again the example given at the beginning of this subsection. Figure 1(c) shows the spectrum estimated with the fast recursive algorithm after the bias was removed with the proposed iterative method. From this example, it's clear that the obtained solution is quite satisfactory.

4. CONCLUSIONS

A significant amount of research attention has been devoted to the estimation of the Capon spectrum. Most of the developed algorithms thus far, however, rely on the direct computation of the inverse of the input correlation (or covariance) matrix. If the length of the Capon filter is L , the complexity of the direct-inverse approach is on the order of L^3 . Such a high computational load makes the Capon algorithm difficult to implement in applications like speech communication where a spectral estimate has to be obtained every few milliseconds. In this paper, we derived a recursive Capon algorithm. This algorithm does not require an explicit matrix inversion, and hence is more efficient to implement than the direct-inverse approach. However, its complexity is still on the order of L^3 . In order to further reduce the complexity and make the recursive Capon algorithm more computationally efficient, a fast version of the recursive algorithm was developed, based on the techniques used in the fast recursive least-squares adaptive algorithms. This new fast algorithm can reduce the complexity by an order of magnitude.

5. REFERENCES

- [1] S. M. Kay, *Modern Spectral Estimation: Theory and Application*. Englewood Cliffs, NJ: Prentice Hall, 1988.
- [2] P. Stoica and R. L. Moses, *Introduction to Spectral Analysis*. Upper Saddle River, NJ: Prentice Hall, 1997.
- [3] J. Capon, "High resolution frequency-wavenumber spectrum analysis," *Proc. IEEE*, vol. 57, pp. 1408–1418, Aug. 1969.
- [4] R. T. Lacoss, "Data adaptive spectral analysis methods," *Geophysics*, vol. 36, pp. 661–675, Aug. 1971.
- [5] M. A. Lagunas, M. E. Santamaria, A. Gasull, and A. Moreno, "Maximum likelihood filters in spectral estimation problems," *Signal Processing*, vol. 10, pp. 19–34, Jan. 1986.
- [6] S. Haykin, *Adaptive Filter Theory*. Fourth Edition, Englewood Cliffs, NJ: Prentice Hall, 2002.
- [7] M. G. Bellanger, *Adaptive Digital Filters and Signal Analysis*. New York: Marcel Dekker, Inc., 1987.