

RESEARCH

Open Access

Adaptive multichannel sequential lattice prediction filtering method for ARMA spectrum estimation in subbands

Mehmet Tahir Ozden

Abstract

A multichannel characterization for autoregressive moving average (ARMA) spectrum estimation in subbands is considered in this article. The fullband ARMA spectrum estimation can be realized in two-channels as a special form of this characterization. A complete orthogonalization of input multichannel data is accomplished using a modified form of sequential processing multichannel lattice stages. Matrix operations are avoided, only scalar operations are used, and a multichannel ARMA prediction filter with a highly modular and suitable structure for VLSI implementations is achieved. Lattice reflection coefficients for autoregressive (AR) and moving average (MA) parts are simultaneously computed. These coefficients are then converted to process parameters using a newly developed Levinson–Durbin type multichannel conversion algorithm. Hence, a novel method for spectrum estimation in subbands as well as in fullband is developed. The computational complexity is given in terms of model order parameters, and comparisons with the complexities of nonparametric methods are provided. In addition, the performance is visually and statistically compared against those of the nonparametric methods under both stationary and nonstationary conditions.

Keywords: Parametric modeling, Subband spectrum estimation and sensing, Frequency estimation and tracking, Radar and speech analysis

1 Introduction

While parametric or model-based methods are used extensively for high-resolution spectrum estimation, these methods perform poorly when SNR and spacing between frequencies is small. In many cases, input noise is assumed to be white; if this is not the case, colored noise can be adapted, provided that its statistics are known. However, such statistics may not be known in many cases, and instead, noise may incorrectly be assumed white. Such shortcomings can be overcome by applying subband decomposition methods in spectrum estimation.

It was shown by Rao and Pearlman [1] that the well-known AR modeling was a promising method for spectrum estimation in subbands, and it was proved that p th-order prediction from subbands is superior to p th-order prediction in the fullband when p is finite, and subband decomposition of a source resulted in a whitening of the composite subband spectrum. The equivalence of linear prediction and AR spectrum estimation was then

exploited to show that AR spectrum from subbands offers a gain over fullband AR spectrum estimation. Unfortunately, new problems such as spectral overlapping and the increase in the variance of estimated parameters appear. The first disadvantage was addressed in a conference paper by Bonacci et al. [2], where nonreal-time procedures have been proposed to perform subband spectral estimation without discontinuities or aliasing at subband borders. However, this procedure is appropriate for a uniform filter bank, even though methods applicable to any kind of filter bank are desired. In another conference paper, Bonacci et al. [3] proposed to tackle the second drawback by a Subband Multichannel Autoregressive Spectral Estimation method, which was also intended for an off-line implementation.

Another popular model, autoregressive moving average (ARMA) model, which includes AR and MA methods as its special cases, has the input–output relationship given by

$$y(n) = - \sum_{\ell=1}^p a_{\ell}^1 y(n - \ell) + \sum_{j=0}^q a_j^2 x(n - j) \quad (1)$$

Correspondence: mtozden@pirireis.edu.tr
Piri Reis University, Tuzla, Istanbul 34940, Turkey

for an ARMA(p, q) process. Here, $x(n)$ is zero mean, white noise with a variance of σ_x^2 , and \hat{a}_ℓ^1 and \hat{a}_j^2 , respectively, represent the ℓ th and j th coefficients related to AR and MA parts. Such processes arise in various applications such as modeling radar signals [4,5] or speech signals [6,7], where spectral zeros as well as poles are often present due to the physical mechanism generating the data. In addition, processes that are purely autoregressive are often transformed into ARMA(p, p) processes by addition of measurement noise, and especially sinusoids in noise are known to obey the degenerate ARMA equation [8,9]. Even though an ARMA process can be represented by a unique AR model of generally infinite order, the ARMA modeling approach often leads to more efficient implementations. A hierarchical ARMA modeling method for classifying high-resolution radar signals at multiple scales was presented in [10], and it was shown that the radar signal at a different scale obeyed an ARMA process if it was an ARMA process at the observed scale.

ARMA model-based applications such as the classification of high-resolution radar signatures using multi-scale features, and lattice speech analysis/synthesis were reported in [11,12], respectively. As a consequence of degenerate ARMA modeling of sinusoids in noise, adaptive multiple frequency tracking, previously considered in [13-15], has gained momentum recently [16], and presents great interest in communications [17], biomedical engineering [18], speech processing [19], and power systems [20,21]. Another recent consequence of degenerate ARMA modeling of sinusoids in noise is related to spectrum sensing for cognitive radios [22,23], where the primary task is to dynamically explore the radio spectrum for the existence of signals (sinusoids) so as to determine portions of the frequency band that may be used for radio transmission. In view of these developments, we think that methods of subband spectrum estimation based on ARMA modeling with possible extensions to full-band spectrum estimation can provide good alternatives in radar and speech classification, adaptive multiple frequency tracking as well as spectrum sensing for cognitive radio applications.

In this article, we propose a novel method that relies on estimation of the driving noise in subbands. Even though methods based on estimation of the driving noise were previously proposed for fullband [24], the important difference of our method is that we first transform the subband ARMA filtering problem into multichannel AR filtering problem by embedding subband ARMA processes into multichannel AR processes, and then we achieve a complete modified Gram-Schmidt orthogonalization of input multichannel signal using a modified version of the sequential processing multichannel lattice stages (SPMLSs) [25]. A number of alternatives for adaptive multichannel processing were proposed after

the introduction of SPMLSs in [25]. Two of such alternatives are the modular lattice architectures proposed by Lev-ari [26], and Glentis and Kalouptsidis [27]. While the architecture in [26] is suitable for equal channel orders and involves more computations than SPMLSs, neither of these architectures is preferable for sequential processing. Another alternative is the QR decomposition-based lattice approach in [28], which is also for equal channel orders, and was later extended to unequal channel orders by Yang [29]. Newer versions of multichannel QR algorithms based on orthogonal Givens rotation for equal as well as unequal channel orders were later presented by Rontogiannis and Theodoridis [30]. Recently, an array-based QR multichannel lattice filter that extends the correspondence between recursive least-squares update equations and Kalman filter equations to the multichannel lattice case was presented by Gomes and Barroso [31]. In addition, transversal-type algorithms such as [32,33] were proposed due to their lower complexity and direct relation to channel coefficients. However, these algorithms generally require the implementation of stabilization techniques, and their structure is less regular. The principle of modular decomposition appears to be the implicit basis in all these adaptive multichannel processing techniques, and provides for the scalar only operations. In QR decomposition approaches, the Q matrix is implicitly formed and then used to compute the R matrix, whereas in the Gram-Schmidt approach, the inverse of the R is implicitly formed and then used to compute the Q matrix. As a consequence of this fact, Regalia and Bellanger [34] showed that there exists a duality between QR and lattice methods, and the possibility of combining elements of both approaches to obtain new hybrid algorithms. With respect to developing these hybrid algorithms, Ling [35] showed that a orthogonal Givens rotation-based algorithm algebraically coincides with the recursive-modified Gram-Schmidt-based lattice algorithm in [36].

In accordance with this perspective in multichannel signal processing, as SPMLSs already have modularity, order recursiveness, regularity, simplicity, sequentiality, and equal as well as unequal channel processing capabilities, we modify them in order to improve their numerical performance by using the error-feedback formula of the recursive-modified Gram-Schmidt algorithm [35,36] in the processing cells. Thus, the complete orthogonalization of multichannel input data and sequential nature of the modified SPMLSs make it possible to feed back the delayed forward prediction error signals to represent the unknown input noise signals of original ARMA processes. Although we introduced the complete orthogonalization concept previously in linear and nonlinear adaptive decision feedback equalization frameworks in [37,38], its application to adaptive spectrum estimation problem in

subbands as well as in fullband results in novel implementations, particularly to the development of a new Levinson–Durbin type conversion algorithm for the modified SPMLSs in order to compute ARMA process parameters from lattice reflection coefficients. To the best of the authors’ knowledge, this particular multichannel lattice prediction filter structure for ARMA spectrum estimation in subbands or in fullband and the new Levinson–Durbin type multichannel conversion algorithm do not exist in the literature.

A two-subband ARMA spectrum estimation problem is considered in this article due to the ease of explanation and space limitations in developing the method. However, it is considered straightforward to apply the method to any number of subbands, and to AR spectrum estimation in subbands. The method is appropriate for uniform and nonuniform filter bank realizations, while aliasing problems due to spectral overlapping in adjacent channels are also addressed. A highly modular, regular, time and order recursive, recursive least squares (RLS) ARMA parameter estimator with inherently good numerical properties, suitable for VLSI and recent programable system on chip implementations [39], is designed, and AR and MA parameters are found simultaneously. With these properties, the method is applicable for both off-line and on-line implementations; it is especially possible to monitor the forward prediction error signal, start the parameter estimation for a fullband AR(p) or ARMA(p, q) or ARMA(p, p) process; if performance requirements are not met, end up for subband ARMA(p_k, q_k) or ARMA(p_k, p_k) realizations. Consequently, it dynamically extends the lattice parametrization of fullband spectrum into subbands, and thereby arises as an useful and practical method for radar signal analysis/classification, speech analysis/synthesis, adaptive multiple frequency tracking, and cognitive radio spectrum sensing tasks.

An adaptive FIR filtering approach to spectral estimation, which is referred to as amplitude and phase estimation of a sinusoid (APES) and has applications to radar target recognition, was proposed by Li and Stoica [40], and the adaptive FIR filtering approach to the Capon method was also discussed by Stoica and Moses [41]. Moreover, the APES method has been extended to array processing by Yardibi et al. [42], and named as iterative adaptive approach for amplitude and phase estimation (IAA-APES). An FIR filtering reinterpretation of the Thomson’s multitaper method [43,44] with applications to spectrum sensing for cognitive radio was also presented by Farhang-Boroujeny [45]. Recently, computationally efficient versions of the adaptive Capon and APES, and IAA methods have been proposed in [46,47], respectively. In this article, we compare the complexity and performance of our method with those of the Periodogram, multitaper, Capon, APES, and IAA methods,

and show that our method is competitive in terms of complexity and performance.

The remainder of this article is organized as follows. In Section 2, we present the development of the new multichannel ARMA lattice prediction filter using the modified SPMLSs. In Section 3, we develop the new Levinson–Durbin type multichannel conversion algorithm for the modified SPMLSs, and relate lattice parameters to process parameters. Spectrum estimation expression in two-subbands is given in Section 4. The computational complexity computations are treated in Section 5. Section 6 is concerned with the experimental results. Finally, Section 7 is about the discussions of results and conclusions. The following notations are used in this article. $(\bullet)^*$ represents the complex conjugate of (\bullet) . $(\bullet)^T$ and $(\bullet)^H$ stand for the transpose and the Hermitian transpose of (\bullet) , respectively. The variables m , i , and n are global while all other variables are local. The variable m represents the stage number while n and i are the time indexes related to data and coefficients, respectively, till we equate them in Section 3 to have a single time index.

2 Adaptive multichannel ARMA lattice prediction filtering

2.1 Multichannel prediction problem

An illustration of the adaptive multichannel ARMA prediction filtering in subbands for two-subband case is presented in Figure 1. Therein, $y(n)$ represents the input fullband signal while $y_1(n)$ and $y_2(n)$ stand for the input subband signals. In adaptive multichannel ARMA prediction filtering, the objective is to find an exponentially windowed, LS solution for the AR and MA coefficients of the k th forward prediction filter that minimizes each of the two cost functions

$$J^k(i) = \sum_{n=0}^i \lambda^{i-n} |f_{p_k}^k(n)|^2 \quad (2)$$

at each time instant i , and $k = 1, 2$. The forward prediction error $f_{p_k}^k(n)$ in this expression is defined as

$$f_{p_k}^k(n) = d^k(n) - \hat{d}_i^k(n) \quad (3)$$

and the k th forward prediction filter output, $\hat{d}_i^k(n)$, is an estimate of the k th desired signal, $d^k(n) = y_k(n)$, is given by

$$\hat{d}_i^k(n) = \sum_{j=1}^{p_k} \tilde{a}_{1,j}^{k*}(i) y_k(n-j) + \sum_{l=0}^{q_k} \tilde{a}_{2,l}^{k*}(i) \hat{u}_k(n-l). \quad (4)$$

Herein, p_k and q_k denote the order of the (p_k, q_k) prediction error filter associated with the k th subband, and $\hat{u}_k(n)$ is the estimate of the k th ARMA process input signal. The estimated k th ARMA process input signal, $\hat{u}_k(n)$, is

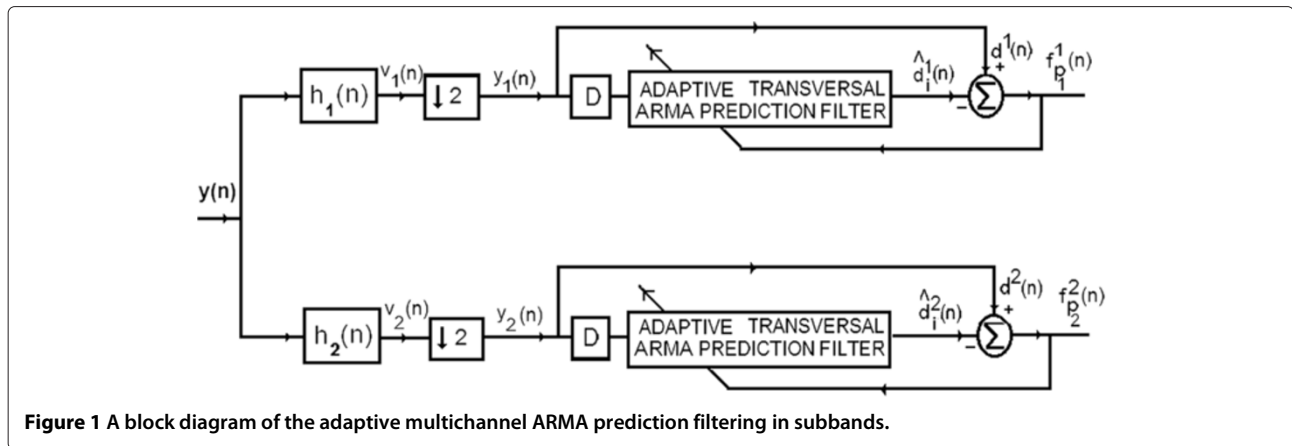


Figure 1 A block diagram of the adaptive multichannel ARMA prediction filtering in subbands.

obtained by delaying and feeding back the p_k th-order forward prediction error, $\hat{u}_k(n) = f_{p_k}^k(n-1)$. Hence, the input vector to the k th ARMA filter at time instant n , $\tilde{\mathbf{y}}_k(n)$, and the corresponding coefficient vector $\tilde{\mathbf{a}}^k(i)$, at time instant i , are defined as

$$\tilde{\mathbf{y}}_k(n) = [y_k(n-1), \dots, y_k(n-p_k), \hat{u}_k(n), \hat{u}_k(n-1), \dots, \hat{u}_k(n-q_k)]^T \quad (5)$$

and

$$\tilde{\mathbf{a}}^{kT}(i) = [\tilde{a}_{1,1}^k(i), \dots, \tilde{a}_{1,p_k}^k(i), \tilde{a}_{2,0}^k(i), \tilde{a}_{2,1}^k(i), \dots, \tilde{a}_{2,q_k}^k(i)], \quad (6)$$

respectively. Herein, $\tilde{a}_{1,j}^k(i)$ and $\tilde{a}_{2,j}^k(i)$, respectively, represent the j th coefficient related to the AR and MA parts of the forward prediction filter for the k th subband at time instant i . It is assumed, without loss of generality, that $p_k \geq q_k$. $p_k = q_k$ case corresponds to the prediction filter for an ARMA(p_k, p_k) process, while $p_k > q_k$ prediction filter is for a general ARMA(p_k, q_k) process. Note that an ARMA backward prediction can be performed for the desired signal, $d^k(n) = y_k(n-p_k)$, and the prediction filter in that case would use the reversed and conjugated forward prediction filter coefficients, which are defined in the backward prediction error coefficient vector as

$$\tilde{\mathbf{c}}^{kT}(i) = [\tilde{c}_{1,p_k}^k(i), \dots, \tilde{c}_{1,1}^k(i), \tilde{c}_{2,q_k}^k(i), \dots, \tilde{c}_{2,1}^k(i), \tilde{c}_{2,0}^k(i)] \quad (7)$$

where $\tilde{c}_{1,j}^k(i)$ and $\tilde{c}_{2,j}^k(i)$ are, respectively, defined as the j th coefficient related to the AR and MA parts of the backward prediction filter for the k th subband at time instant i .

Consequently, the main concern of the exponentially weighted LS problem under consideration is to find, at

each time i , the k th optimal coefficient vector, $\tilde{\mathbf{a}}^k(i)$ that would minimize the cost function

$$J^k(i) = \sum_{n=0}^i \lambda^{i-n} |d^k(n) - \tilde{\mathbf{a}}^{kH}(i)\tilde{\mathbf{y}}_k(n)|^2. \quad (8)$$

The k th optimal coefficient vector related to the k th subband filter

$$\tilde{\mathbf{a}}_{opt}^k(i) = \mathbf{R}_k^{-1}(i)\mathbf{P}_k(i) \quad (9)$$

is found by differentiating $J^k(i)$ with respect to $\tilde{\mathbf{a}}^k(i)$, setting the derivative to zero, and solving for $\tilde{\mathbf{a}}^k(i)$, where

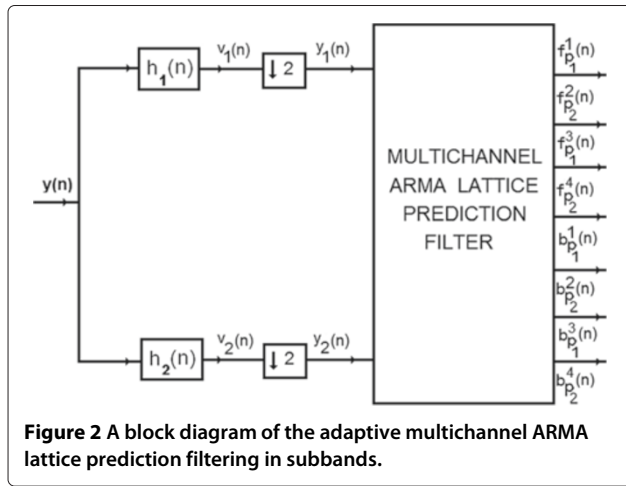
$$\mathbf{R}_k(i) = \sum_{n=0}^i \lambda^{i-n} \tilde{\mathbf{y}}_k(n)\tilde{\mathbf{y}}_k^H(n) \quad (10)$$

and

$$\mathbf{P}_k(i) = \sum_{n=0}^i \lambda^{i-n} \tilde{\mathbf{y}}_k(n)d^{k*}(n). \quad (11)$$

2.2 Sequential lattice orthogonalization

In order to find a modular, regular, and simple solution to the two-subband ARMA prediction problem, we would like to use a single multichannel lattice filter as depicted in Figure 2, instead of using two separate transversal filters and solving two separate optimization problems as in Figure 1. We would also like to avoid direct evaluations as in (9), and achieve good numerical properties. As the number of channels at different sections of the proposed multichannel lattice filter is different due to the sequential processing nature of SPMLs, we carry out the exponentially weighted LS optimization problem by taking into consideration each of these sections separately, and therefore we assume that the filter is comprised of three cascaded filters, which are two-channel, three-channel, and four-channel lattice sections; and we use a different index for each section while using m to indicate a stage in the whole filter. We also assume $p_1 = p_2$ for the ease of explanation without loss of generality.



In order to sequentially solve the exponentially weighted *LS* optimization problem under consideration, we first organize the elements of input signal vectors $\mathbf{y}_1(n) = [y_1(n), \dots, y_1(n-\ell)]^T$, and $\mathbf{y}_2(n) = [y_2(n), \dots, y_2(n-\ell)]^T$ according to the natural ordering of SPMLSs as

$$\bar{\mathbf{y}}_{\ell+1}(n) = \begin{bmatrix} y_1(n) \\ y_2(n) \\ y_1(n-1) \\ y_2(n-1) \\ \dots \\ y_1(n-\ell) \\ y_2(n-\ell) \end{bmatrix} \quad (12)$$

and input to two-channel stages for which the stage number (m) has a range of values given by $0 < m \leq (p_1 - q_1)$. Accordingly, we redefine Equations (10) and (11) using this new data vector as follows

$$\mathbf{R}_\ell(i) = \sum_{n=0}^i \lambda^{i-n} \bar{\mathbf{y}}_{\ell+1}(n) \bar{\mathbf{y}}_{\ell+1}^H(n) \quad (13)$$

and

$$\mathbf{P}_{\ell,k}(i) = \sum_{n=0}^i \lambda^{i-n} \bar{\mathbf{y}}_{\ell+1}(n) d^{k*}(n) \quad (14)$$

where $k = 1, 2$. The orthogonalization of data using SPMLSs corresponds to the transformation of (13) and (14) into

$$\mathbf{D}_{\ell+1}^f(i) = \sum_{n=0}^i \lambda^{i-n} \mathbf{\Omega}_\ell^f(i) \bar{\mathbf{y}}_{\ell+1}(n) \bar{\mathbf{y}}_{\ell+1}^H(n) \mathbf{\Omega}_\ell^{fH}(i) \quad (15)$$

and

$$\mathbf{Z}_{\ell+1,k}^f(i) = \sum_{n=0}^i \lambda^{i-n} \mathbf{\Omega}_\ell^f(i-1) \bar{\mathbf{y}}_{\ell+1}(n-1) d^{k*}(n), \quad (16)$$

respectively. Here, $\mathbf{\Omega}_\ell^f(i)$ is the $2\ell \times 2\ell$ lower triangular transformation matrix for forward prediction, and is sequentially realized stage-by-stage using 2×2 lower triangular transformation matrices

$$\mathbf{L}_\ell^f(i) = \begin{bmatrix} 1 & 0 \\ \hat{\kappa}_\ell^f(i-1) & 1 \end{bmatrix} \quad (17)$$

whose diagonal elements are all equal to unity at time instant i , and $\hat{\kappa}_\ell^f(i)$ is the reflection coefficient computed at the single circular cell in the triangular-shaped self-orthogonalization processor of the ℓ th two-channel SPMLS. Then, the forward lattice predictor coefficients are computed using

$$\mathbf{\Theta}_{\ell,k}^f(i) = \mathbf{D}_{\ell+1}^{-f}(i-1) \mathbf{Z}_{\ell+1,k}^f(i) \quad (18)$$

where $\mathbf{\Theta}_{\ell,k}^f(i)$ represents the k th row of the $2 \times 2\ell$ lattice forward prediction reflection coefficient matrix $\mathbf{\Theta}_\ell^f(i)$, and is also sequentially implemented stage-by-stage by means of 2×2 forward prediction reflection coefficient matrices

$$\mathbf{\Delta}_\ell^f(i) = \begin{bmatrix} \bar{\kappa}_{\ell,1,1}^f(i) & \bar{\kappa}_{\ell,1,2}^f(i) \\ \bar{\kappa}_{\ell,2,1}^f(i) & \bar{\kappa}_{\ell,2,2}^f(i) \end{bmatrix} \quad (19)$$

in which $\bar{\kappa}_{\ell,k,j}^f(i)$ is the j th reflection coefficient related to the forward prediction of the k th channel signal, and it is computed at the (k, j) th single circular cell of the square-shaped reference-orthogonalization processor related to forward prediction at the ℓ th two-channel SPMLS. Note that the matrix inversion operation in Equation (9) is transformed into a simple scalar inversion operation in (18) due to the diagonal nature of $\mathbf{D}_{\ell+1}^f(i)$. The backward prediction counterpart of this optimization problem is similarly solved using 2×2 lower triangular transformation matrices $\mathbf{L}_\ell^b(i)$, and 2×2 lattice backward prediction reflection coefficient matrices, $\mathbf{\Delta}_\ell^b(i)$.

After the processing of input signals by two-channel lattice stages, the delayed and fed back forward prediction error $\hat{u}_1(n) = f_{p_1}(n-1)$ is incorporated at the $(p_1 - q_1 + 1)$ th stage, as the third channel. Accordingly, we expand the optimization problem by organizing the elements of the input data vectors $\mathbf{y}_1(n) = [y_1(n), \dots, y_1(n-\alpha)]^T$, $\mathbf{y}_2(n) = [y_2(n), \dots, y_2(n-\alpha)]^T$, and $\hat{\mathbf{u}}_1(n) = [\hat{u}_1(n), \dots, \hat{u}_1(n-\alpha)]^T$ as follows:

$$\bar{\mathbf{y}}_{\alpha+1}(n) = \begin{bmatrix} y_1(n) \\ y_2(n) \\ \hat{u}_1(n) \\ \dots \\ y_1(n-\alpha) \\ y_2(n-\alpha) \\ \hat{u}_1(n-\alpha) \end{bmatrix}, \quad (20)$$

and input to three-channel lattice section, where the stage number (m) takes values in the range given by $(p_1 - q_1) < m \leq (p_2 - q_2)$. Subsequently, we solve the optimization problem in (18) once again with the new input vector, in which case $\mathbf{\Omega}_\alpha^f(i)$ and $\mathbf{\Theta}_\alpha^f(i)$ are the $3\alpha \times 3\alpha$ lower triangular transformation and the $3 \times 3\alpha$ forward lattice prediction coefficient matrices, respectively. $\mathbf{\Omega}_\alpha^f(i)$ is computed sequentially by means of 3×3 lower triangular transformation matrices, $\mathbf{L}_\alpha^f(i)$, and $\mathbf{\Theta}_\alpha^f(i)$ is similarly realized stage-by-stage making use of 3×3 forward prediction coefficient matrices, $\mathbf{\Delta}_\alpha^f(i)$, at time instant i . Note that, since the delayed and fed back signal is considered to constitute a new channel in the multichannel sequential lattice filtering, we have three desired signals at this point, $d^k(n)$, where $k = 1, 2, 3$, one of which did not exist in the optimization problem stated in Section 2.1, and this new desired signal, $d^3(n)$, is related to the MA part of the first subband ARMA modeling.

Finally, the optimization problem is expanded one more time with the inclusion of the second delayed and fed back forward prediction error $\hat{u}_2(n) = f_{p_2}(n-1)$, and this time, the elements of input data vectors $\mathbf{y}_1(n) = [y_1(n), \dots, y_1(n-\nu)]^T$, $\mathbf{y}_2(n) = [y_2(n), \dots, y_2(n-\nu)]^T$, $\hat{\mathbf{u}}_1(n) = [\hat{u}_1(n), \dots, \hat{u}_1(n-\nu)]^T$, and $\hat{\mathbf{u}}_2(n) = [\hat{u}_2(n), \dots, \hat{u}_2(n-\nu)]^T$ are organized as

$$\bar{\mathbf{y}}_{v+1}(n) = \begin{bmatrix} y_1(n) \\ y_2(n) \\ \hat{u}_1(n) \\ \hat{u}_2(n) \\ - \\ - \\ y_1(n-\nu) \\ y_2(n-\nu) \\ \hat{u}_1(n-\nu) \\ \hat{u}_2(n-\nu) \end{bmatrix} \quad (21)$$

where the stage number (m) is in the range given by $(p_2 - q_2) < m \leq p_2$ due to four-channel processing. Similar to two-channel and three-channel cases, we solve the optimization problem in (18) using the new data vector in Equation (21), in which case $\mathbf{\Omega}_v^f(i)$ and $\mathbf{\Theta}_v^f(i)$ are $4\nu \times 4\nu$ lower triangular transformation, and $4 \times 4\nu$ forward lattice prediction coefficient matrices at the time instant i , respectively. Similar to previous cases, these matrices are computed stage-by-stage by the use of 4×4 lower triangular transformation matrices, $\mathbf{L}_v^f(i)$, and 4×4 forward prediction coefficient matrices, $\mathbf{\Delta}_v^f(i)$, at time instant i , respectively. As the second delayed and fed back signal is also considered as a new channel in the multichannel sequential lattice filtering, hereafter we have four desired signals, $d^k(n)$, where $k = 1, 2, 3, 4$, and this fourth desired signal, $d^4(n)$, is associated with the MA part of the second subband ARMA modeling.

2.3 Matrix visualization

In order to further explain the sequential lattice orthogonalization, we consider a (8, 5) and (8, 2) ARMA prediction lattice prediction filter for the first and second subbands, and organize the elements of input data vectors $\mathbf{y}_1(n) = [y_1(n), \dots, y_1(n-8)]^T$, $\mathbf{y}_2(n) = [y_2(n), \dots, y_2(n-8)]^T$, $\hat{\mathbf{u}}_1(n) = [\hat{u}_1(n), \hat{u}_1(n-1), \dots, \hat{u}_1(n-5)]^T$, and $\hat{\mathbf{u}}_2(n) = [\hat{u}_2(n), \hat{u}_2(n-1), \dots, \hat{u}_2(n-2)]^T$ as columns of a matrix,

$$\begin{bmatrix} y_1(n) & y_1(n-1) & y_1(n-2) & y_1(n-3) & y_1(n-4) & y_1(n-5) & y_1(n-6) & y_1(n-7) & y_1(n-8) \\ y_2(n) & y_2(n-1) & y_2(n-2) & y_2(n-3) & y_2(n-4) & y_2(n-5) & y_2(n-6) & y_2(n-7) & y_2(n-8) \\ & & \hat{u}_1(n) & \hat{u}_1(n-1) & \hat{u}_1(n-2) & \hat{u}_1(n-3) & \hat{u}_1(n-4) & \hat{u}_1(n-5) & \\ & & & & \hat{u}_2(n) & \hat{u}_2(n-1) & \hat{u}_2(n-2) & & \end{bmatrix} \quad (22)$$

by taking into consideration different number of parameters in the feedforward and feedback channels and shifting properties of input data. This matrix helps us to visualize the orthogonalization process, and thus to draw a diagram of the four-channel prediction filter structure under consideration as in Figure 3. Note that the elements of the first and second rows are related to the input signals of the first and the second subband channels of the ARMA filter under consideration, while the third and fourth rows are associated with the fed back and delayed signals. Lattice orthogonalization begins with the elements of the first two rows using two-channel sequential lattice processing stages until the first fed back and delayed channel is incorporated as the new channel. Then, the orthogonalization continues with three-channel lattice stages until the fourth channel, which is the second fed back and delayed channel, is taken into the process, and so the orthogonalization of input data finalizes with four-channel stages when the mean squared prediction error performance requirements are met, and thereby the k th desired signal, $d^k(n)$, is sequentially predicted using self-orthogonalized and delayed backward prediction error signals as follows:

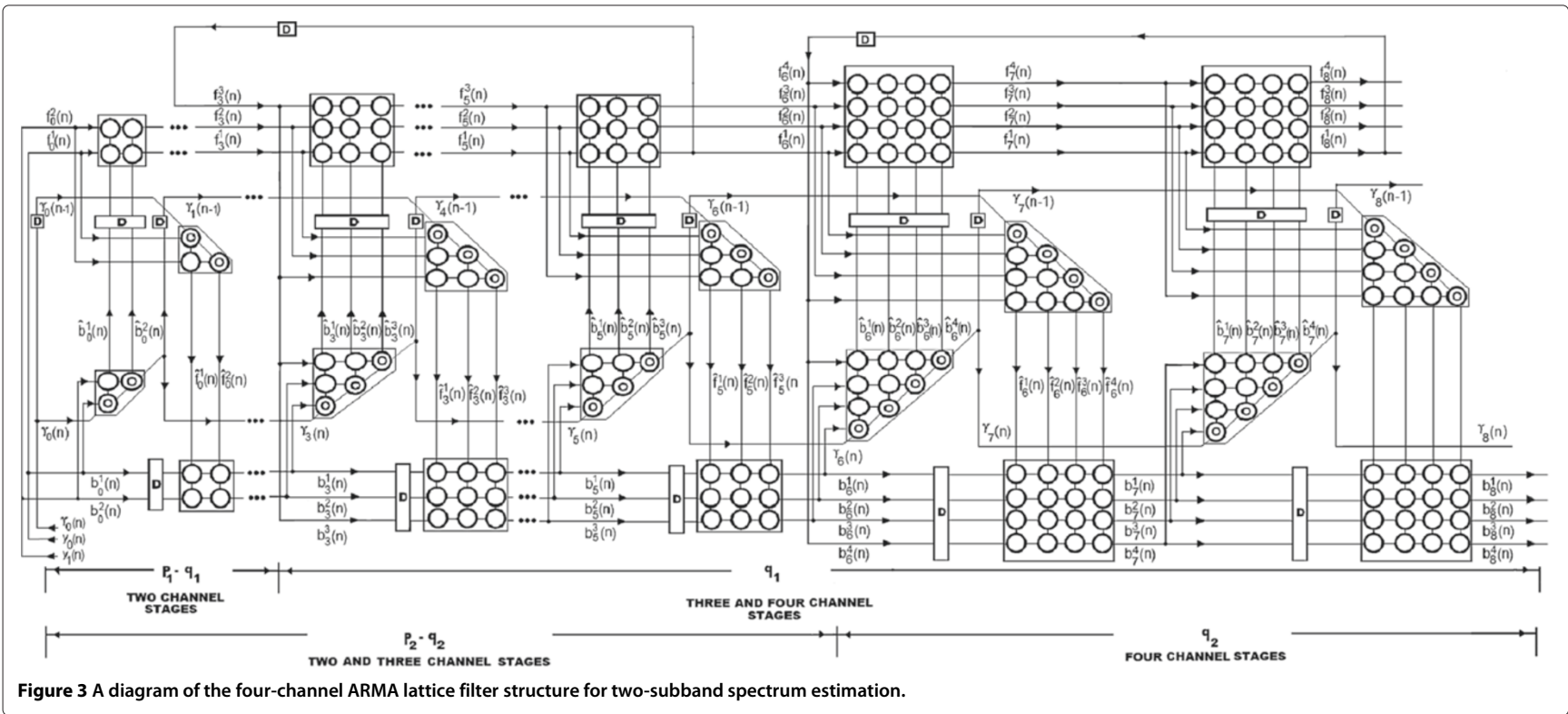


Figure 3 A diagram of the four-channel ARMA lattice filter structure for two-subband spectrum estimation.

$$\begin{aligned} \hat{d}_i^k(n) = & \sum_{m=1}^{p_1-q_1} \sum_{j=1}^2 \bar{\kappa}_{m,k,j}^{f*}(i-1) \hat{b}_{m-1}^j(n-1) \\ & + \sum_{m=p_1-q_1+1}^{p_2-q_2} \sum_{j=1}^3 \bar{\kappa}_{m,k,j}^{f*}(i-1) \hat{b}_{m-1}^j(n-1) \\ & + \sum_{m=p_2-q_2+1}^{p_2} \sum_{j=1}^4 \bar{\kappa}_{m,k,j}^{f*}(i-1) \hat{b}_{m-1}^j(n-1). \end{aligned} \quad (23)$$

Here, the first and second summations represent the prediction accomplished by the two-channel and three-channel sections, respectively, and the fourth summation is connected with the four-channel prediction section. In each section, $\bar{\kappa}_{m,k,j}^f(i)$ represents the j th forward prediction reflection coefficient at the m th stage related to the k th channel as defined in the previous subsection, and $\hat{b}_{m-1}^j(n)$ represents the j th element of the self-orthogonalized backward prediction error signal vector, $\hat{\mathbf{b}}_{m-1}(n)$, at the input of the m th stage. The self-orthogonalized backward prediction error vector, $\hat{\mathbf{b}}_{m-1}(n)$, is produced by the lower triangular transformation of the input backward prediction error vector, $\mathbf{b}_{m-1}(n)$, using $\mathbf{L}_m^f(n)$, and this operation is accomplished at the triangular shaped self-orthogonalization processor (related to forward prediction) of the m th SPMLS. Note that the sizes of vectors, $\hat{\mathbf{b}}_{m-1}(n)$, $\mathbf{b}_{m-1}(n)$, and matrix, $\mathbf{L}_m^f(n)$, at different sections of the proposed lattice filter are as follows: 2×1 , and 2×2 in two-channel section, 3×1 , and 3×3 in three-channel section, and 4×1 , and 4×4 in four-channel section, respectively.

We would also like to point out that a lattice filter for fullband ARMA spectrum estimation is a special form of the two-subband implementation, and therefore it can similarly be realized using sequential processing one-channel and two-channel lattice stages as illustrated in Figure 4 for an ARMA(10,2) implementation.

3 Conversion of lattice coefficients to process parameters

Since the mathematical link between process parameters and reflection coefficients of a lattice prediction filter is provided by the Levinson–Durbin algorithm [48,49], we develop a new Levinson–Durbin type conversion algorithm specifically for SPMLSs in order to convert lattice reflection coefficients to subband ARMA process parameters. Due to the sequential nature of the proposed lattice structure, we carry out the development of the new Levinson–Durbin type multichannel conversion algorithm by taking into consideration each of these sections separately, and therefore we assume that the filter is comprised of three cascaded filters as in Section 2.2.

We first consider the conversion algorithm for the two-channel section of lattice prediction filter, and we organize the input signal samples to two-channel lattices as

$$\bar{\mathbf{y}}_{\ell+1}(n) = \begin{bmatrix} \mathbf{y}_1(n) \\ \text{-----} \\ \mathbf{y}_1(n-\ell) \\ \mathbf{y}_2(n) \\ \text{-----} \\ \mathbf{y}_2(n-\ell) \end{bmatrix} = \begin{bmatrix} \mathbf{y}_1(n) \\ \text{-----} \\ \mathbf{y}_1(n-1) \\ \mathbf{y}_2(n) \\ \text{-----} \\ \mathbf{y}_2(n-1) \end{bmatrix} \quad (24)$$

where we define the data vectors as $\mathbf{y}_1(n) = [y_1(n), \dots, y_1(n-\ell+1)]^T$, $\mathbf{y}_2(n) = [y_2(n), \dots, y_2(n-\ell+1)]^T$, and $0 < m \leq (p_1 - q_1)$. The corresponding forward and backward prediction error coefficient matrices for the ℓ th-order transversal filter for the k th channel are defined as

$$\hat{\mathbf{a}}^{kT}(i) = [\hat{a}_0^k(i), \hat{a}_1^k(i), \hat{a}_2^k(i), \dots, \hat{a}_{2\ell-2}^k(i), \hat{a}_{2\ell-1}^k(i), \hat{a}_{2\ell}^k(i)] \quad (25)$$

and

$$\hat{\mathbf{c}}^{kT}(i) = [\hat{c}_{2\ell}^k(i), \hat{c}_{2\ell-1}^k(i), \hat{c}_{2\ell-2}^k(i), \dots, \hat{c}_2^k(i), \hat{c}_1^k(i), \hat{c}_0^k(i)] \quad (26)$$

where $k = 1, 2$ due to two-channel lattice processing, and $\hat{a}_0^k(i) = \hat{c}_0^k(i) = 1.0$. Since the signal time shifting and ordering properties of SPMLSs when expressed in matrix form as in Equation (12) are different than the organization of input signal samples in matrix form as in Equation (24), we use $(2\ell + 1) \times (2\ell + 1)$ shuffling matrices, $\mathbf{J}_{\ell+1}^1$ for the first channel and $\mathbf{J}_{\ell+1}^2$ for the second channel, to reorder the elements of coefficient matrices, $\hat{\mathbf{a}}_\ell^{1H}(i)$, $\hat{\mathbf{a}}_\ell^{2H}(i)$ and $\hat{\mathbf{c}}_\ell^{1H}(i)$, $\hat{\mathbf{c}}_\ell^{2H}(i)$, according to the sample ordering of SPMLSs. Therefore, the forward and backward prediction errors for the end of the observation interval $n = i$ at the output of the general ℓ th-order filters with transversal structure can be stated as

$$\begin{bmatrix} f_\ell^1(n) \\ f_\ell^2(n) \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{\ell+1}^1 \hat{\mathbf{a}}_\ell^{1H}(n) & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{\ell+1}^2 \hat{\mathbf{a}}_\ell^{2H}(n) \end{bmatrix} \bar{\mathbf{y}}_{\ell+1}(n) \quad (27)$$

$$\begin{bmatrix} b_\ell^1(n) \\ b_\ell^2(n) \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{\ell+1}^1 \hat{\mathbf{c}}_\ell^{1H}(n) & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{\ell+1}^2 \hat{\mathbf{c}}_\ell^{2H}(n) \end{bmatrix} \bar{\mathbf{y}}_{\ell+1}(n) \quad (28)$$

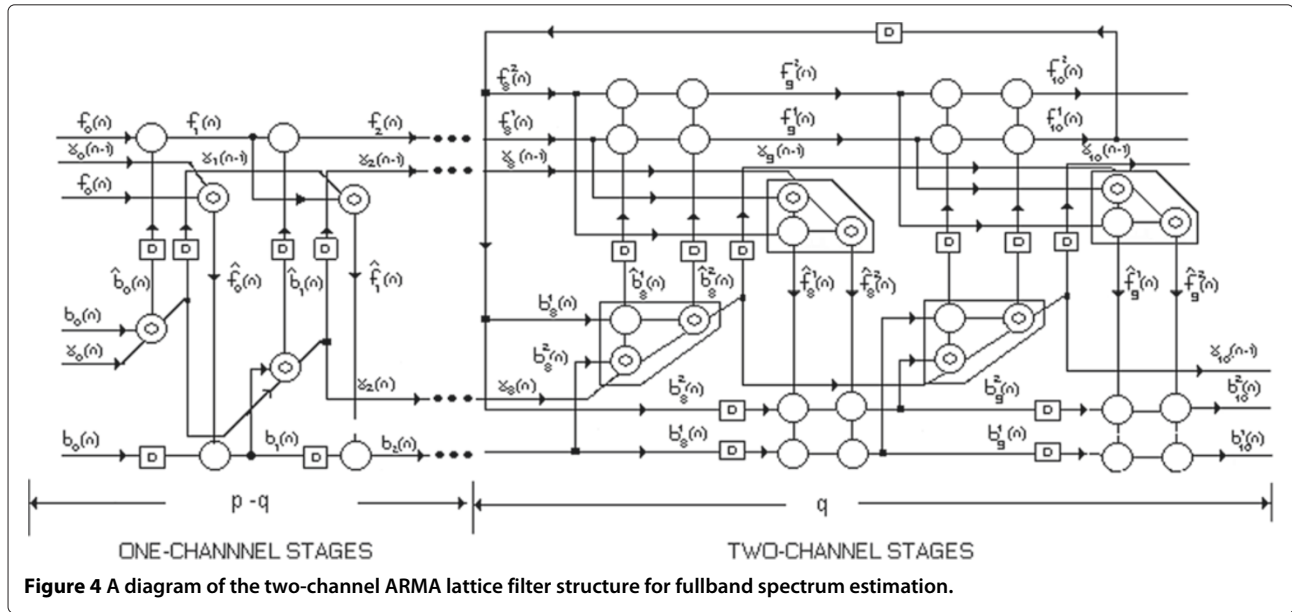


Figure 4 A diagram of the two-channel ARMA lattice filter structure for fullband spectrum estimation.

where $\mathbf{0}$ is a $1 \times (\ell + 1)$ zero matrix. Then, we can express the $(\ell - 1)$ th prediction errors as

$$\begin{bmatrix} f_{\ell-1}^1(n) \\ f_{\ell-1}^2(n) \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{\ell+1}^1 [\hat{\mathbf{a}}_{\ell-1}^{1H}(n) \ 0 \ 0] & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{\ell+1}^2 [\hat{\mathbf{a}}_{\ell-1}^{2H}(n) \ 0 \ 0] \end{bmatrix} \tilde{\mathbf{y}}_{\ell+1}(n) \quad (29)$$

$$\begin{bmatrix} b_{\ell-1}^1(n-1) \\ b_{\ell-1}^2(n-1) \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{\ell+1}^1 [0 \ 0 \ \hat{\mathbf{c}}_{\ell-1}^{1H}(n-1)] & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{\ell+1}^2 [0 \ 0 \ \hat{\mathbf{c}}_{\ell-1}^{2H}(n-1)] \end{bmatrix} \tilde{\mathbf{y}}_{\ell+1}(n) \quad (30)$$

Note that the size of each coefficient matrix increases by two when the order of prediction filter increases from $\ell - 1$ to ℓ , and $\mathbf{0}$ is a $1 \times (\ell + 1)$ zero matrix as before. Subsequently, we define the ℓ th-order prediction errors in terms of lattice parameters and the $(\ell - 1)$ th-order prediction errors as follows

$$\begin{bmatrix} f_{\ell}^1(n) \\ f_{\ell}^2(n) \end{bmatrix} = \begin{bmatrix} f_{\ell-1}^1(n) \\ f_{\ell-1}^2(n) \end{bmatrix} + \begin{bmatrix} \bar{\kappa}_{\ell,1,1}^{f*}(n-1) & \bar{\kappa}_{\ell,1,2}^{f*}(n-1) \\ \bar{\kappa}_{\ell,2,1}^{f*}(n-1) & \bar{\kappa}_{\ell,2,2}^{f*}(n-1) \end{bmatrix} \times \begin{bmatrix} 1 & 0 \\ \hat{\kappa}_{\ell}^{f*}(n-2) & 1 \end{bmatrix} \begin{bmatrix} b_{\ell-1}^1(n-1) \\ b_{\ell-1}^2(n-1) \end{bmatrix} \quad (31)$$

$$\begin{bmatrix} b_{\ell}^1(n) \\ b_{\ell}^2(n) \end{bmatrix} = \begin{bmatrix} b_{\ell-1}^1(n-1) \\ b_{\ell-1}^2(n-1) \end{bmatrix} + \begin{bmatrix} \bar{\kappa}_{\ell,1,1}^{b*}(n-1) & \bar{\kappa}_{\ell,1,2}^{b*}(n-1) \\ \bar{\kappa}_{\ell,2,1}^{b*}(n-1) & \bar{\kappa}_{\ell,2,2}^{b*}(n-1) \end{bmatrix} \times \begin{bmatrix} 1 & 0 \\ \hat{\kappa}_{\ell}^{b*}(n-1) & 1 \end{bmatrix} \begin{bmatrix} f_{\ell-1}^1(n) \\ f_{\ell-1}^2(n) \end{bmatrix} \quad (32)$$

where the lower coefficient triangular and square matrices are generated in triangular shaped self-orthogonalization and square shaped reference-orthogonalization processors in a two-channel SPMLS as defined in Equations (17) and (19). Accordingly, we multiply these lower triangular and square coefficient matrices, and make the following definitions

$$\begin{aligned} \Gamma_{\ell}^f(n) &= \begin{bmatrix} \Gamma_{\ell,1,1}^f(n) & \Gamma_{\ell,1,2}^f(n) \\ \Gamma_{\ell,2,1}^f(n) & \Gamma_{\ell,2,2}^f(n) \end{bmatrix} \\ &= \begin{bmatrix} \bar{\kappa}_{\ell,1,1}^f(n) + \bar{\kappa}_{\ell,1,2}^f(n) \hat{\kappa}_{\ell}^f(n-1) & \bar{\kappa}_{\ell,1,2}^f(n) \\ \bar{\kappa}_{\ell,2,1}^f(n) + \bar{\kappa}_{\ell,2,2}^f(n) \hat{\kappa}_{\ell}^f(n-1) & \bar{\kappa}_{\ell,2,2}^f(n) \end{bmatrix} \end{aligned} \quad (33)$$

$$\begin{aligned} \Gamma_{\ell}^b(n) &= \begin{bmatrix} \Gamma_{\ell,1,1}^b(n) & \Gamma_{\ell,1,2}^b(n) \\ \Gamma_{\ell,2,1}^b(n) & \Gamma_{\ell,2,2}^b(n) \end{bmatrix} \\ &= \begin{bmatrix} \bar{\kappa}_{\ell,1,1}^b(n) + \bar{\kappa}_{\ell,1,2}^b(n) \hat{\kappa}_{\ell}^b(n) & \bar{\kappa}_{\ell,1,2}^b(n) \\ \bar{\kappa}_{\ell,2,1}^b(n) + \bar{\kappa}_{\ell,2,2}^b(n) \hat{\kappa}_{\ell}^b(n) & \bar{\kappa}_{\ell,2,2}^b(n) \end{bmatrix} \end{aligned} \quad (34)$$

in order to obtain compact versions of Equations (31) and (32) as follows

$$\begin{bmatrix} f_{\ell}^1(n) \\ f_{\ell}^2(n) \end{bmatrix} = \begin{bmatrix} f_{\ell-1}^1(n) \\ f_{\ell-1}^2(n) \end{bmatrix} + \Gamma_{\ell}^{f*}(n-1) \begin{bmatrix} b_{\ell-1}^1(n-1) \\ b_{\ell-1}^2(n-1) \end{bmatrix} \quad (35)$$

$$\begin{bmatrix} b_\ell^1(n) \\ b_\ell^2(n) \end{bmatrix} = \begin{bmatrix} b_{\ell-1}^1(n-1) \\ b_{\ell-1}^2(n-1) \end{bmatrix} + \Gamma_\ell^{b*}(n-1) \begin{bmatrix} f_{\ell-1}^1(n) \\ f_{\ell-1}^2(n) \end{bmatrix}. \quad (36)$$

Then, the ℓ th-order prediction error matrices in Equations (27) and (28), and the $(\ell-1)$ th-order prediction error matrices in Equations (29) and (30) are substituted in the ℓ th-order prediction error expressions in (35) and (36) so as to obtain the following pairs of order updates

$$\begin{aligned} \hat{\mathbf{a}}_\ell^1(n) &= \begin{bmatrix} \hat{\mathbf{a}}_{\ell-1}^1(n) \\ \mathbf{0} \end{bmatrix} + \Gamma_{\ell,1,1}^{f*}(n-1) \mathbf{J}_{\ell+1}^1 \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{c}}_{\ell-1}^1(n-1) \end{bmatrix} \\ &+ \Gamma_{\ell,1,2}^{f*}(n-1) \mathbf{J}_{\ell+1}^2 \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{c}}_{\ell-1}^2(n-1) \end{bmatrix} \end{aligned} \quad (37)$$

$$\begin{aligned} \hat{\mathbf{a}}_\ell^2(n) &= \begin{bmatrix} \hat{\mathbf{a}}_{\ell-1}^2(n) \\ \mathbf{0} \end{bmatrix} + \Gamma_{\ell,2,1}^{f*}(n-1) \mathbf{J}_{\ell+1}^1 \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{c}}_{\ell-1}^1(n-1) \end{bmatrix} \\ &+ \Gamma_{\ell,2,2}^{f*}(n-1) \mathbf{J}_{\ell+1}^2 \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{c}}_{\ell-1}^2(n-1) \end{bmatrix} \end{aligned} \quad (38)$$

$$\begin{aligned} \hat{\mathbf{c}}_\ell^1(n) &= \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{c}}_{\ell-1}^1(n-1) \end{bmatrix} + \Gamma_{\ell,1,1}^{b*}(n-1) \mathbf{J}_{\ell+1}^1 \begin{bmatrix} \hat{\mathbf{a}}_{\ell-1}^1(n) \\ \mathbf{0} \end{bmatrix} \\ &+ \Gamma_{\ell,1,2}^{b*}(n-1) \mathbf{J}_{\ell+1}^2 \begin{bmatrix} \hat{\mathbf{a}}_{\ell-1}^2(n) \\ \mathbf{0} \end{bmatrix} \end{aligned} \quad (39)$$

$$\begin{aligned} \hat{\mathbf{c}}_\ell^2(n) &= \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{c}}_{\ell-1}^2(n-1) \end{bmatrix} + \Gamma_{\ell,2,1}^{b*}(n-1) \mathbf{J}_{\ell+1}^1 \begin{bmatrix} \hat{\mathbf{a}}_{\ell-1}^1(n) \\ \mathbf{0} \end{bmatrix} \\ &+ \Gamma_{\ell,2,2}^{b*}(n-1) \mathbf{J}_{\ell+1}^2 \begin{bmatrix} \hat{\mathbf{a}}_{\ell-1}^2(n) \\ \mathbf{0} \end{bmatrix} \end{aligned} \quad (40)$$

and since the size of each coefficient matrix increase by two, $\mathbf{0}$ is a 2×1 zero matrix. The three-channel section starts with the incorporation of the third channel ($\hat{u}_1(n)$) as the new channel at the $(p_1 - q_1 + 1)$ th stage. In order to develop the Levinson–Durbin algorithm for this section, we assume that three-channel section is a separate filter, and thereby considering the input signal samples to the three-channel section as follows

$$\bar{\mathbf{y}}_{\alpha+1}(n) = \begin{bmatrix} \begin{bmatrix} \mathbf{y}_1(n) \\ \text{---} \\ y_1(n-\alpha) \end{bmatrix} \\ \begin{bmatrix} \mathbf{y}_2(n) \\ \text{---} \\ y_2(n-\alpha) \end{bmatrix} \\ \begin{bmatrix} \hat{\mathbf{u}}_1(n) \\ \text{---} \\ \hat{u}_1(n-\alpha) \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} y_1(n) \\ \text{---} \\ y_1(n-1) \end{bmatrix} \\ \begin{bmatrix} y_2(n) \\ \text{---} \\ y_2(n-1) \end{bmatrix} \\ \begin{bmatrix} \hat{u}_1(n) \\ \text{---} \\ \hat{u}_1(n-1) \end{bmatrix} \end{bmatrix} \quad (41)$$

where $\mathbf{y}_1(n) = [y_1(n), \dots, y_1(n-\alpha+1)]^T$, $\mathbf{y}_2(n) = [y_2(n), \dots, y_2(n-\alpha+1)]^T$, and $\hat{\mathbf{u}}_1(n) = [\hat{u}_1(n), \dots, \hat{u}_1(n-\alpha+1)]^T$. Correspondingly, the forward and backward prediction error coefficient matrices for the α th-order transversal filtering are defined as

$$\begin{aligned} \hat{\mathbf{a}}^k(n) &= [\hat{a}_0^k(n), \hat{a}_1^k(n), \hat{a}_2^k(n), \hat{a}_3^k(n), \dots, \\ &\hat{a}_{3\alpha-2}^k(n), \hat{a}_{3\alpha-1}^k(n), \hat{a}_{3\alpha}^k(n)] \end{aligned} \quad (42)$$

and

$$\begin{aligned} \hat{\mathbf{c}}^k(n) &= [\hat{c}_{3\alpha}^k(n), \hat{c}_{3\alpha-1}^k(n), \hat{c}_{3\alpha-2}^k(n), \dots, \\ &\hat{c}_3^k(n), \hat{c}_2^k(n), \hat{c}_1^k(n), \hat{c}_0^k(n)] \end{aligned} \quad (43)$$

where $k = 1, 2, 3$ due to three-channel processing. Then, the prediction filtering continues with three-channel lattice stages for $(p_1 - q_1) < m \leq (p_2 - q_2)$. The Levinson–Durbin recursions for the three-channel section can be developed similar to the two-channel section by establishing the mathematical link between transversal and lattice filter coefficients. Since the organization of signal samples in Equation (41) is different than the ordering of signal samples entering into three-channel SPMLSs in (20), we use $(3\alpha + 1) \times (3\alpha + 1)$ shuffling matrices, $\mathbf{J}_{\alpha+1}^1$ for the first channel, $\mathbf{J}_{\alpha+1}^2$ for the second channel, and $\mathbf{J}_{\alpha+1}^3$ for the third channel to reorder the elements of coefficient matrices, $\hat{\mathbf{a}}_\alpha^{1H}(n)$, $\hat{\mathbf{a}}_\alpha^{2H}(n)$, $\hat{\mathbf{a}}_\alpha^{3H}(n)$ and $\hat{\mathbf{c}}_\alpha^{1H}(n)$, $\hat{\mathbf{c}}_\alpha^{2H}(n)$, $\hat{\mathbf{c}}_\alpha^{3H}(n)$, according to the sample ordering of SPMLSs. Similar to Equations (27) and (28) in two-channel case, the forward and backward prediction errors in three-channel case for the output of the general α th-order filter with transversal structure are expressed as

$$\begin{bmatrix} f_\alpha^1(n) \\ f_\alpha^2(n) \\ f_\alpha^3(n) \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{\alpha+1}^1 \hat{\mathbf{a}}_\alpha^{1H}(n) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{\alpha+1}^2 \hat{\mathbf{a}}_\alpha^{2H}(n) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{J}_{\alpha+1}^3 \hat{\mathbf{a}}_\alpha^{3H}(n) \end{bmatrix} \bar{\mathbf{y}}_{\alpha+1}(n) \quad (44)$$

$$\begin{bmatrix} b_\alpha^1(n) \\ b_\alpha^2(n) \\ b_\alpha^3(n) \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{\alpha+1}^1 \hat{\mathbf{c}}_\alpha^{1H}(n) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{\alpha+1}^2 \hat{\mathbf{c}}_\alpha^{2H}(n) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{J}_{\alpha+1}^3 \hat{\mathbf{c}}_\alpha^{3H}(n) \end{bmatrix} \bar{\mathbf{y}}_{\alpha+1}(n) \quad (45)$$

where $\mathbf{0}$ is a $1 \times (\alpha + 1)$ zero matrix in this case. We can then express the $(\alpha - 1)$ th prediction errors as

$$\begin{bmatrix} f_{\alpha-1}^1(n) \\ f_{\alpha-1}^2(n) \\ f_{\alpha-1}^3(n) \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{\alpha+1}^1 [\hat{\mathbf{a}}_{\alpha-1}^{1H}(n) \ 0 \ 0 \ 0] & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{\alpha+1}^2 [\hat{\mathbf{a}}_{\alpha-1}^{2H}(n) \ 0 \ 0 \ 0] & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{J}_{\alpha+1}^3 [\hat{\mathbf{a}}_{\alpha-1}^{3H}(n) \ 0 \ 0 \ 0] \end{bmatrix} \bar{\mathbf{y}}_{\alpha+1}(n) \quad (46)$$

$$\begin{bmatrix} b_{\alpha-1}^1(n-1) \\ b_{\alpha-1}^2(n-1) \\ b_{\alpha-1}^3(n-1) \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{\alpha+1}^1 [0 \ 0 \ 0 \ \hat{\mathbf{c}}_{\alpha-1}^{1H}(n-1)] & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{\alpha+1}^2 [0 \ 0 \ 0 \ \hat{\mathbf{c}}_{\alpha-1}^{2H}(n-1)] & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{J}_{\alpha+1}^3 [0 \ 0 \ 0 \ \hat{\mathbf{c}}_{\alpha-1}^{3H}(n-1)] \end{bmatrix} \bar{\mathbf{y}}_{\alpha+1}(n). \quad (47)$$

Note that the size of each coefficient matrix in three-channel case increases by three when the order of prediction filter increases from $\alpha - 1$ to α . Similar to Equations (35) and (36) in two-channel case, the lattice prediction errors for the α th three-channel stage can be expressed in compact form with the following equations

$$\begin{bmatrix} f_{\alpha}^1(n) \\ f_{\alpha}^2(n) \\ f_{\alpha}^3(n) \end{bmatrix} = \begin{bmatrix} f_{\alpha-1}^1(n) \\ f_{\alpha-1}^2(n) \\ f_{\alpha-1}^3(n) \end{bmatrix} + \Gamma_{\alpha}^{f*}(n-1) \begin{bmatrix} b_{\alpha-1}^1(n-1) \\ b_{\alpha-1}^2(n-1) \\ b_{\alpha-1}^3(n-1) \end{bmatrix} \quad (48)$$

$$\begin{bmatrix} b_{\alpha}^1(n) \\ b_{\alpha}^2(n) \\ b_{\alpha}^3(n) \end{bmatrix} = \begin{bmatrix} b_{\alpha-1}^1(n-1) \\ b_{\alpha-1}^2(n-1) \\ b_{\alpha-1}^3(n-1) \end{bmatrix} + \Gamma_{\alpha}^{b*}(n-1) \begin{bmatrix} f_{\alpha-1}^1(n) \\ f_{\alpha-1}^2(n) \\ f_{\alpha-1}^3(n) \end{bmatrix} \quad (49)$$

where

$$\begin{aligned} \Gamma_{\alpha}^f(n) &= \begin{bmatrix} \Gamma_{\alpha,1,1}^f(n) & \Gamma_{\alpha,1,2}^f(n) & \Gamma_{\alpha,1,3}^f(n) \\ \Gamma_{\alpha,2,1}^f(n) & \Gamma_{\alpha,2,2}^f(n) & \Gamma_{\alpha,2,3}^f(n) \\ \Gamma_{\alpha,3,1}^f(n) & \Gamma_{\alpha,3,2}^f(n) & \Gamma_{\alpha,3,3}^f(n) \end{bmatrix} \\ &= \begin{bmatrix} \bar{\kappa}_{\alpha,1,1}^f(n) & \bar{\kappa}_{\alpha,1,2}^f(n) & \bar{\kappa}_{\alpha,1,3}^f(n) \\ \bar{\kappa}_{\alpha,2,1}^f(n) & \bar{\kappa}_{\alpha,2,2}^f(n) & \bar{\kappa}_{\alpha,2,3}^f(n) \\ \bar{\kappa}_{\alpha,3,1}^f(n) & \bar{\kappa}_{\alpha,3,2}^f(n) & \bar{\kappa}_{\alpha,3,3}^f(n) \end{bmatrix} \\ &\quad \times \begin{bmatrix} 1 & 0 & 0 \\ \hat{\kappa}_{\alpha,2,1}^f(n-1) & 1 & 0 \\ \hat{\kappa}_{\alpha,3,1}^f(n-1) & \hat{\kappa}_{\alpha,3,2}^f(n-1) & 1 \end{bmatrix} \end{aligned}$$

and

$$\begin{aligned} \Gamma_{\alpha}^b(n) &= \begin{bmatrix} \Gamma_{\alpha,1,1}^b(n) & \Gamma_{\alpha,1,2}^b(n) & \Gamma_{\alpha,1,3}^b(n) \\ \Gamma_{\alpha,2,1}^b(n) & \Gamma_{\alpha,2,2}^b(n) & \Gamma_{\alpha,2,3}^b(n) \\ \Gamma_{\alpha,3,1}^b(n) & \Gamma_{\alpha,3,2}^b(n) & \Gamma_{\alpha,3,3}^b(n) \end{bmatrix} \\ &= \begin{bmatrix} \bar{\kappa}_{\alpha,1,1}^b(n) & \bar{\kappa}_{\alpha,1,2}^b(n) & \bar{\kappa}_{\alpha,1,3}^b(n) \\ \bar{\kappa}_{\alpha,2,1}^b(n) & \bar{\kappa}_{\alpha,2,2}^b(n) & \bar{\kappa}_{\alpha,2,3}^b(n) \\ \bar{\kappa}_{\alpha,3,1}^b(n) & \bar{\kappa}_{\alpha,3,2}^b(n) & \bar{\kappa}_{\alpha,3,3}^b(n) \end{bmatrix} \\ &\quad \times \begin{bmatrix} 1 & 0 & 0 \\ \hat{\kappa}_{\alpha,2,1}^b(n) & 1 & 0 \\ \hat{\kappa}_{\alpha,3,1}^b(n) & \hat{\kappa}_{\alpha,3,2}^b(n) & 1 \end{bmatrix}. \end{aligned}$$

The α th-order prediction error matrices in Equations (44) and (45), and the $(\alpha - 1)$ th-order prediction error matrices in Equations (46) and (47) are subsequently substituted in the α th-order prediction error expressions in (48) and (49) so that the following pairs of order updates are produced

$$\begin{aligned} \hat{\mathbf{a}}_{\alpha}^1(n) &= \begin{bmatrix} \hat{\mathbf{a}}_{\alpha-1}^1(n) \\ \mathbf{0} \end{bmatrix} + \Gamma_{\alpha,1,1}^{f*}(n-1) \mathbf{J}_{\alpha+1}^1 \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{c}}_{\alpha-1}^1(n-1) \end{bmatrix} \\ &\quad + \dots + \Gamma_{\alpha,1,3}^{f*}(n-1) \mathbf{J}_{\alpha+1}^3 \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{c}}_{\alpha-1}^3(n-1) \end{bmatrix} \end{aligned} \quad (50)$$

$$\begin{aligned} \hat{\mathbf{a}}_{\alpha}^2(n) &= \begin{bmatrix} \hat{\mathbf{a}}_{\alpha-1}^2(n) \\ \mathbf{0} \end{bmatrix} + \Gamma_{\alpha,2,1}^{f*}(n-1) \mathbf{J}_{\alpha+1}^1 \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{c}}_{\alpha-1}^1(n-1) \end{bmatrix} \\ &\quad + \dots + \Gamma_{\alpha,2,3}^{f*}(n-1) \mathbf{J}_{\alpha+1}^3 \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{c}}_{\alpha-1}^3(n-1) \end{bmatrix} \end{aligned} \quad (51)$$

$$\begin{aligned} \hat{\mathbf{a}}_{\alpha}^3(n) &= \begin{bmatrix} \hat{\mathbf{a}}_{\alpha-1}^3(n) \\ \mathbf{0} \end{bmatrix} + \Gamma_{\alpha,3,1}^{f*}(n-1) \mathbf{J}_{\alpha+1}^1 \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{c}}_{\alpha-1}^1(n-1) \end{bmatrix} \\ &\quad + \dots + \Gamma_{\alpha,3,3}^{f*}(n-1) \mathbf{J}_{\alpha+1}^3 \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{c}}_{\alpha-1}^3(n-1) \end{bmatrix} \end{aligned} \quad (52)$$

$$\begin{aligned} \hat{\mathbf{c}}_{\alpha}^1(n) &= \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{c}}_{\alpha-1}^1(n-1) \end{bmatrix} + \Gamma_{\alpha,1,1}^{b*}(n-1) \mathbf{J}_{\alpha+1}^1 \begin{bmatrix} \hat{\mathbf{a}}_{\alpha-1}^1(n) \\ \mathbf{0} \end{bmatrix} \\ &\quad + \dots + \Gamma_{\alpha,1,3}^{b*}(n-1) \mathbf{J}_{\alpha+1}^3 \begin{bmatrix} \hat{\mathbf{a}}_{\alpha-1}^3(n) \\ \mathbf{0} \end{bmatrix} \end{aligned} \quad (53)$$

$$\begin{aligned} \hat{\mathbf{c}}_{\alpha}^2(n) &= \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{c}}_{\alpha-1}^2(n-1) \end{bmatrix} + \Gamma_{\alpha,2,1}^{b*}(n-1) \mathbf{J}_{\alpha+1}^1 \begin{bmatrix} \hat{\mathbf{a}}_{\alpha-1}^1(n) \\ \mathbf{0} \end{bmatrix} \\ &\quad + \dots + \Gamma_{\alpha,2,3}^{b*}(n-1) \mathbf{J}_{\alpha+1}^3 \begin{bmatrix} \hat{\mathbf{a}}_{\alpha-1}^3(n) \\ \mathbf{0} \end{bmatrix} \end{aligned} \quad (54)$$

$$\begin{aligned} \hat{\mathbf{c}}_{\alpha}^3(n) &= \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{c}}_{\alpha-1}^3(n-1) \end{bmatrix} + \Gamma_{\alpha,3,1}^{b*}(n-1) \mathbf{J}_{\alpha+1}^1 \begin{bmatrix} \hat{\mathbf{a}}_{\alpha-1}^1(n) \\ \mathbf{0} \end{bmatrix} \\ &\quad + \dots + \Gamma_{\alpha,3,3}^{b*}(n-1) \mathbf{J}_{\alpha+1}^3 \begin{bmatrix} \hat{\mathbf{a}}_{\alpha-1}^3(n) \\ \mathbf{0} \end{bmatrix} \end{aligned} \quad (55)$$

where the size of $\mathbf{0}$ is a 3×1 zero matrix. Finally, the fourth channel ($\hat{u}_2(n)$), which represents the fed back and delayed signal related to the second subband, is taken into the orthogonalization process at the $(p_2 - q_2 + 1)$ th stage, and the prediction filtering continues with four-channel lattice stages through $(p_2 - q_2) < m \leq p_2$. In order to develop the Levinson–Durbin recursions for this section, we define the forward and backward prediction error coefficient matrices for the ν th-order transversal filtering as

$$\hat{\mathbf{a}}^{kT}(n) = \left[\hat{a}_0^k(n), \hat{a}_1^k(n), \hat{a}_2^k(n), \hat{a}_3^k(n), \hat{a}_4^k(n), \dots, \dots, \right. \\ \left. \hat{a}_{4\nu-3}^k(n), \hat{a}_{4\nu-2}^k(n), \hat{a}_{4\nu-1}^k(n), \hat{a}_{4\nu}^k(n) \right] \quad (56)$$

and

$$\hat{\mathbf{c}}^{kT}(n) = \left[\hat{c}_{4\nu}^k(n), \hat{c}_{4\nu-1}^k(n), \hat{c}_{4\nu-2}^k(n), \hat{c}_{4\nu-3}^k(n), \dots, \dots, \right. \\ \left. \hat{c}_4^k(n), \hat{c}_3^k(n), \hat{c}_2^k(n), \hat{c}_1^k(n), \hat{c}_0^k(n) \right] \quad (57)$$

where $k = 1, 2, 3, 4$ due to four-channel lattice processing, and we also visualize as before that the following organization of the elements of input vectors $\mathbf{y}_1(n) = [y_1(n), \dots, y_1(n - \nu + 1)]^T$, $\mathbf{y}_2(n) = [y_2(n), \dots, y_2(n - \nu + 1)]^T$, $\hat{\mathbf{u}}_1(n) = [\hat{u}_1(n), \dots, \hat{u}_1(n - \nu + 1)]^T$, and $\hat{\mathbf{u}}_2(n) = [\hat{u}_2(n), \dots, \hat{u}_2(n - \nu + 1)]^T$ is established:

$$\bar{\mathbf{y}}_{\nu+1}(n) = \begin{bmatrix} \begin{bmatrix} \mathbf{y}_1(n) \\ \text{-----} \\ y_1(n - \nu) \end{bmatrix} \\ \begin{bmatrix} \mathbf{y}_2(n) \\ \text{-----} \\ y_2(n - \nu) \end{bmatrix} \\ \begin{bmatrix} \hat{\mathbf{u}}_1(n) \\ \text{-----} \\ \hat{u}_1(n - \nu) \end{bmatrix} \\ \begin{bmatrix} \hat{\mathbf{u}}_2(n) \\ \text{-----} \\ \hat{u}_2(n - \nu) \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} y_1(n) \\ \text{-----} \\ \mathbf{y}_1(n - 1) \end{bmatrix} \\ \begin{bmatrix} y_2(n) \\ \text{-----} \\ \mathbf{y}_2(n - 1) \end{bmatrix} \\ \begin{bmatrix} \hat{u}_1(n) \\ \text{-----} \\ \hat{\mathbf{u}}_1(n - 1) \end{bmatrix} \\ \begin{bmatrix} \hat{u}_2(n) \\ \text{-----} \\ \hat{\mathbf{u}}_2(n - 1) \end{bmatrix} \end{bmatrix}. \quad (58)$$

Similar to the previous two steps, the signal sample ordering in Equation (58) is different than the ordering in Equation (21), hence we use $(4\nu + 1) \times (4\nu + 1)$ shuffling matrices, $\mathbf{J}_{\nu+1}^1$ for the first channel, $\mathbf{J}_{\nu+1}^2$ for the second channel, $\mathbf{J}_{\nu+1}^3$ for the third channel, and $\mathbf{J}_{\nu+1}^4$ for the fourth channel to reorder the elements of coefficient matrices $\hat{\mathbf{a}}_{\nu}^{1H}(n)$, $\hat{\mathbf{a}}_{\nu}^{2H}(n)$, $\hat{\mathbf{a}}_{\nu}^{3H}(n)$, $\hat{\mathbf{a}}_{\nu}^{4H}(n)$ and $\hat{\mathbf{c}}_{\nu}^{1H}(n)$, $\hat{\mathbf{c}}_{\nu}^{2H}(n)$, $\hat{\mathbf{c}}_{\nu}^{3H}(n)$, $\hat{\mathbf{c}}_{\nu}^{4H}(n)$, according to the sample

ordering of SPMLSs. Then, the development of the Levinson–Durbin recursions for this section unfolds as in two and three-channel sections. First, the ν th and the $(\nu - 1)$ th order forward and backward prediction errors are stated as the output of a transversal. Second, the prediction order update equations for the $(\nu - 1)$ th and the ν th-orders are expressed for a four-channel lattice section, and finally the ν th and the $(\nu - 1)$ th-order forward and backward transversal filter prediction error expressions are substituted in the lattice prediction order update equations such that the following pairs of order updates are obtained

$$\hat{\mathbf{a}}_{\nu}^1(n) = \begin{bmatrix} \hat{\mathbf{a}}_{\nu-1}^1(n) \\ \mathbf{0} \end{bmatrix} + \Gamma_{\nu,1,1}^{f*}(n-1)\mathbf{J}_{\nu+1}^1 \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{c}}_{\nu-1}^1(n-1) \end{bmatrix} \\ + \dots + \Gamma_{\nu,1,4}^{f*}(n-1)\mathbf{J}_{\nu+1}^4 \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{c}}_{\nu-1}^4(n-1) \end{bmatrix} \quad (59)$$

$$\hat{\mathbf{a}}_{\nu}^2(n) = \begin{bmatrix} \hat{\mathbf{a}}_{\nu-1}^2(n) \\ \mathbf{0} \end{bmatrix} + \Gamma_{\nu,2,1}^{f*}(n-1)\mathbf{J}_{\nu+1}^1 \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{c}}_{\nu-1}^1(n-1) \end{bmatrix} \\ + \dots + \Gamma_{\nu,2,4}^{f*}(n-1)\mathbf{J}_{\nu+1}^4 \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{c}}_{\nu-1}^4(n-1) \end{bmatrix} \quad (60)$$

$$\hat{\mathbf{a}}_{\nu}^3(n) = \begin{bmatrix} \hat{\mathbf{a}}_{\nu-1}^3(n) \\ \mathbf{0} \end{bmatrix} + \Gamma_{\nu,3,1}^{f*}(n-1)\mathbf{J}_{\nu+1}^1 \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{c}}_{\nu-1}^1(n-1) \end{bmatrix} \\ + \dots + \Gamma_{\nu,3,4}^{f*}(n-1)\mathbf{J}_{\nu+1}^4 \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{c}}_{\nu-1}^4(n-1) \end{bmatrix} \quad (61)$$

$$\hat{\mathbf{a}}_{\nu}^4(n) = \begin{bmatrix} \hat{\mathbf{a}}_{\nu-1}^4(n) \\ \mathbf{0} \end{bmatrix} + \Gamma_{\nu,4,1}^{f*}(n-1)\mathbf{J}_{\nu+1}^1 \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{c}}_{\nu-1}^1(n-1) \end{bmatrix} \\ + \dots + \Gamma_{\nu,4,4}^{f*}(n-1)\mathbf{J}_{\nu+1}^4 \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{c}}_{\nu-1}^4(n-1) \end{bmatrix} \quad (62)$$

$$\hat{\mathbf{c}}_{\nu}^1(n) = \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{c}}_{\nu-1}^1(n-1) \end{bmatrix} + \Gamma_{\nu,1,1}^{b*}(n-1)\mathbf{J}_{\nu+1}^1 \begin{bmatrix} \hat{\mathbf{a}}_{\nu-1}^1(n) \\ \mathbf{0} \end{bmatrix} \\ + \dots + \Gamma_{\nu,1,4}^{b*}(n-1)\mathbf{J}_{\nu+1}^4 \begin{bmatrix} \hat{\mathbf{a}}_{\nu-1}^4(n) \\ \mathbf{0} \end{bmatrix} \quad (63)$$

$$\hat{\mathbf{c}}_{\nu}^2(n) = \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{c}}_{\nu-1}^2(n-1) \end{bmatrix} + \Gamma_{\nu,2,1}^{b*}(n-1)\mathbf{J}_{\nu+1}^1 \begin{bmatrix} \hat{\mathbf{a}}_{\nu-1}^1(n) \\ \mathbf{0} \end{bmatrix} \\ + \dots + \Gamma_{\nu,2,4}^{b*}(n-1)\mathbf{J}_{\nu+1}^4 \begin{bmatrix} \hat{\mathbf{a}}_{\nu-1}^4(n) \\ \mathbf{0} \end{bmatrix} \quad (64)$$

$$\begin{aligned} \hat{\mathbf{c}}_v^3(n) &= \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{c}}_{v-1}^3(n-1) \end{bmatrix} + \Gamma_{v,3,1}^{b*}(n-1) \mathbf{J}_{v+1}^1 \begin{bmatrix} \hat{\mathbf{a}}_{v-1}^1(n) \\ \mathbf{0} \end{bmatrix} \\ &+ \dots + \Gamma_{v,3,4}^{b*}(n-1) \mathbf{J}_{v+1}^4 \begin{bmatrix} \hat{\mathbf{a}}_{v-1}^4(n) \\ \mathbf{0} \end{bmatrix} \end{aligned} \quad (65)$$

$$\begin{aligned} \hat{\mathbf{c}}_v^4(n) &= \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{c}}_{v-1}^4(n-1) \end{bmatrix} + \Gamma_{v,4,1}^{b*}(n-1) \mathbf{J}_{v+1}^1 \begin{bmatrix} \hat{\mathbf{a}}_{v-1}^1(n) \\ \mathbf{0} \end{bmatrix} \\ &+ \dots + \Gamma_{v,4,4}^{b*}(n-1) \mathbf{J}_{v+1}^4 \begin{bmatrix} \hat{\mathbf{a}}_{v-1}^4(n) \\ \mathbf{0} \end{bmatrix}. \end{aligned} \quad (66)$$

Note that $\mathbf{0}$ is a 4×1 zero matrix, and that conversion of lattice parameters to process parameters started with two channels, but ended with four channels due to sequential processing. The new Levinson–Durbin type conversion algorithm for a fullband ARMA spectrum estimation can be similarly developed as a special case of subband implementation. The lattice prediction filter for fullband ARMA spectrum estimation, which consists of one and two-channel sections, is shown in Figure 4. The corresponding conversion algorithm can also be realized in two sections as summarized in Subsection New Levinson-Durbin Type Conversion Algorithm for Two-Channel ARMA Lattice Prediction.

3.1 New Levinson-Durbin type conversion algorithm for two-channel ARMA lattice prediction

Initialization :

$$\begin{aligned} \hat{a}_0^1(n) &= 1.0, \quad \hat{c}_0^1(n) = 1.0, \quad \hat{a}_{p-q}^2(n) = 1.0, \\ \hat{c}_0^2(n) &= 1.0. \end{aligned} \quad (67)$$

One-channel Section ($0 < m \leq (p - q)$) :

$$\hat{\mathbf{a}}_m^1(n) = \begin{bmatrix} \hat{\mathbf{a}}_{m-1}^1(n) \\ 0 \end{bmatrix} + \Gamma_{m,1}^{f*}(n-1) \begin{bmatrix} 0 \\ \hat{\mathbf{c}}_{m-1}^1(n-1) \end{bmatrix} \quad (68)$$

$$\hat{\mathbf{c}}_m^1(n) = \begin{bmatrix} 0 \\ \hat{\mathbf{c}}_{m-1}^1(n-1) \end{bmatrix} + \Gamma_{m,1}^{b*}(n-1) \begin{bmatrix} \hat{\mathbf{a}}_{m-1}^1(n) \\ 0 \end{bmatrix}. \quad (69)$$

Two-channel Section ($(p - q) < m \leq p$) :

$$\begin{aligned} \hat{\mathbf{a}}_m^1(n) &= \begin{bmatrix} \hat{\mathbf{a}}_{m-1}^1(n) \\ 0 \end{bmatrix} + \Gamma_{m,1,1}^{f*}(n-1) \mathbf{J}_{m+1}^1 \begin{bmatrix} 0 \\ \hat{\mathbf{c}}_{m-1}^1(n-1) \end{bmatrix} \\ &+ \Gamma_{m,1,2}^{f*}(n-1) \mathbf{J}_{m+1}^2 \begin{bmatrix} 0 \\ \hat{\mathbf{c}}_{m-1}^2(n-1) \end{bmatrix} \end{aligned} \quad (70)$$

$$\begin{aligned} \hat{\mathbf{a}}_m^2(n) &= \begin{bmatrix} \hat{\mathbf{a}}_{m-1}^2(n) \\ 0 \end{bmatrix} + \Gamma_{m,2,1}^{f*}(n-1) \mathbf{J}_{m+1}^1 \begin{bmatrix} 0 \\ \hat{\mathbf{c}}_{m-1}^1(n-1) \end{bmatrix} \\ &+ \Gamma_{m,2,2}^{f*}(n-1) \mathbf{J}_{m+1}^2 \begin{bmatrix} 0 \\ \hat{\mathbf{c}}_{m-1}^2(n-1) \end{bmatrix} \end{aligned} \quad (71)$$

$$\begin{aligned} \hat{\mathbf{c}}_m^1(n) &= \begin{bmatrix} 0 \\ \hat{\mathbf{c}}_{m-1}^1(n-1) \end{bmatrix} + \Gamma_{m,1,1}^{b*}(n-1) \mathbf{J}_{m+1}^1 \begin{bmatrix} \hat{\mathbf{a}}_{m-1}^1(n) \\ 0 \end{bmatrix} \\ &+ \Gamma_{m,1,2}^{b*}(n-1) \mathbf{J}_{m+1}^2 \begin{bmatrix} \hat{\mathbf{a}}_{m-1}^2(n) \\ 0 \end{bmatrix} \end{aligned} \quad (72)$$

$$\begin{aligned} \hat{\mathbf{c}}_m^2(n) &= \begin{bmatrix} 0 \\ \hat{\mathbf{c}}_{m-1}^2(n-1) \end{bmatrix} + \Gamma_{m,2,1}^{b*}(n-1) \mathbf{J}_{m+1}^1 \begin{bmatrix} \hat{\mathbf{a}}_{m-1}^1(n) \\ 0 \end{bmatrix} \\ &+ \Gamma_{m,2,2}^{b*}(n-1) \mathbf{J}_{m+1}^2 \begin{bmatrix} \hat{\mathbf{a}}_{m-1}^2(n) \\ 0 \end{bmatrix}. \end{aligned} \quad (73)$$

4 Spectrum estimation from subbands

After computing process parameters from lattice coefficients, we established the link between multichannel prediction and spectrum estimation. Hence, the estimated spectrum in subbands can be expressed in terms of the subband prediction filter parameters as

$$\begin{aligned} S_y(w_k) &= \left| \frac{1 + \bar{a}_1^3 e^{-jw_k} + \dots + \bar{a}_{q_1}^3 e^{-jq_1 w_k}}{1 + \bar{a}_1^1 e^{-jw_k} + \dots + \bar{a}_{p_1}^1 e^{-jp_1 w_k}} \right|^2 \cdot \hat{\sigma}_{x_1}^2 \\ &+ \left| \frac{1 + \bar{a}_1^4 e^{-jw_k} + \dots + \bar{a}_{q_2}^4 e^{-jq_2 w_k}}{1 + \bar{a}_1^2 e^{-jw_k} + \dots + \bar{a}_{p_2}^2 e^{-jp_2 w_k}} \right|^2 \cdot \hat{\sigma}_{x_2}^2 \end{aligned} \quad (74)$$

where $\hat{\sigma}_{x_k}^2$ represents the prediction error variance for the k th subband; and the coefficients, $\bar{a}_1^1, \dots, \bar{a}_{p_1}^1$ and $\bar{a}_1^3, \dots, \bar{a}_{q_1}^3$ are related to the AR and MA parts of the first subband ARMA spectrum while the coefficients $\bar{a}_1^2, \dots, \bar{a}_{p_2}^2$ and $\bar{a}_1^4, \dots, \bar{a}_{q_2}^4$ are associated with AR and MA parts of the second subband ARMA spectrum. Specifically, we determine the coefficients related to the first and second subbands in Equation (74) from the elements of coefficient vectors in Equations (25), (42), and (56) using the coefficient selection rule given in Subsection Coefficient Selection Rule for Process Parameters in Four-Channel ARMA Lattice Prediction. Note that we omit the extra coefficients, \hat{a}_0^1 and \hat{a}_0^2 , in Equation (42), and \hat{a}_0^1 , \hat{a}_0^2 , and \hat{a}_0^3 in Equation (56) as they had appeared due to separate filter assumption for the sections of ARMA lattice prediction filter. We also present the coefficient selection rule for the two-channel fullband case in Subsection Coefficient Selection Rule for Process Parameters in Two-Channel ARMA Lattice Prediction.

4.1 Coefficient selection rule for process parameters in four-channel ARMA lattice prediction

Two-channel section ($1 \leq \ell \leq (p_1 - q_1)$):

$$\bar{a}_\ell^1(n) = \hat{a}_{2\ell-1}^1(n) \quad (75)$$

$$\bar{a}_\ell^2(n) = \hat{a}_{2\ell}^2(n) \quad (76)$$

Three-channel section ($1 \leq \ell \leq (p_1 - q_1) - (p_2 - q_2)$):

$$\bar{a}_{\ell+p_1-q_1}^1(n) = \hat{a}_{3\ell-2}^1(n) \quad (77)$$

$$\bar{a}_{\ell+p_1-q_1}^2(n) = \hat{a}_{3\ell-1}^2(n) \quad (78)$$

$$\bar{a}_{\ell+p_1-q_1}^3(n) = \hat{a}_{3\ell}^3(n) \quad (79)$$

Four-channel section ($1 \leq \ell \leq q_2$):

$$\bar{a}_{\ell+p_2-q_2}^1(n) = \hat{a}_{4\ell-3}^1(n) \quad (80)$$

$$\bar{a}_{\ell+p_2-q_2}^2(n) = \hat{a}_{4\ell-2}^2(n) \quad (81)$$

$$\bar{a}_{\ell+p_2-q_2}^3(n) = \hat{a}_{4\ell-1}^3(n) \quad (82)$$

$$\bar{a}_{\ell+p_2-q_2}^4(n) = \hat{a}_{4\ell}^4(n) \quad (83)$$

4.2 Coefficient selection rule for process parameters in two-channel ARMA lattice prediction

One-channel section ($1 \leq \ell \leq (p - q)$):

$$\bar{a}_\ell^1(n) = \hat{a}_\ell^1(n) \quad (84)$$

Two-channel section ($1 \leq \ell \leq q$):

$$\bar{a}_{\ell+p-q}^1(n) = \hat{a}_{2\ell-1}^1(n) \quad (85)$$

$$\bar{a}_{\ell+p-q}^2(n) = \hat{a}_{2\ell}^2(n) \quad (86)$$

The connection between subband and fullband frequencies can be explained using an example from [15]. Accordingly, a sinusoid of frequency w_0 in fullband is mapped into the frequency w_M in subbands with

$$w_M = M.w_0 \text{ mod}(2\pi) \quad (87)$$

where M is the number of subbands. On the other hand, knowing the sinusoid frequency w_M at subbands, the frequency w_0 can be obtained by

$$w_0 = \frac{2\pi}{M}.K + \frac{w_M}{M} \quad (88)$$

where K is the integer part of $\frac{M.w_0}{2\pi}$.

5 Computational complexity

The number of operations required for the two-channel ARMA lattice prediction filter for fullband spectrum estimation is calculated as $10p + 16q$ using the number of operations required for one-channel and two-channel sequential processing lattice stages [25], where ‘‘one operation is considered as one multiplication (division) and one addition’’. The Levinson–Durbin recursion for one-channel lattice sections requires $(p - q)(p - q + 1)$ operations, and $4q(q + 1)$ operations for two-channel lattice sections to compute the ARMA process parameters. Therefore, the number of operations required becomes $p^2 + 11p - 2pq + 5q^2 + 19q$, and then this expression can be extended to the total number of required operations for an M subband, multichannel implementation as

$$\text{Total complexity} = \sum_{k=1}^M p_k^2 + 11p_k - 2p_k q_k + 5q_k^2 + 19q_k. \quad (89)$$

Accordingly, we would like to compare the total number of operations for the proposed method with adaptive transversal filtering, and the nonparametric methods, namely, the Periodogram, multitaper, Capon, APES, and IAA methods.

The computational complexity of a fast RLS transversal ARMA filter can also be expressed in order of p_k and q_k [50]. When the fast Fourier transform (FFT) is utilized in implementing the Periodogram method, the required number of operations, which is the total number of real additions(subtractions) and multiplications(divisions) is $C_{\text{FFT}}(N) = 4N \log_2 N$, where N is the number of signal samples and is a power of 2 [51].

The computational complexity of the multitaper method is then approximately given by $C_{MT} \approx NW C_{\text{FFT}}(N)$, where NW and $2W$ are defined as the time-bandwidth product and the resolution bandwidth, respectively [41,43]. The complexity of brute force computations of the adaptive Capon and APES spectral estimators are given in [46] as $C_{\text{CAPON}}(N_f, K) \approx N_f^3 + N_f^2 K$ and $C_{\text{APES}}(N_f, L_w, K) \approx N_f^3 + N_f^2 L_w^2 + (N_f^2 + L_w^2 + N_f L_w)$, respectively, where K represents the size of uniformly spaced grid of frequencies, N_f is the filter length, and L_w is the sliding window size. It is also shown in [46] that these complexities can be reduced to $C_{\text{CAPON}}(K) \approx 12K$ and $C_{\text{APES}}(K) \approx 42K$ if computationally efficient versions of adaptive Capon and APES spectral estimators, which are classified as FRLS-III type, are utilized. Similarly, the complexity of brute force version of the IAA spectral estimator is provided in [47] as $C_{\text{IAA}} = m_c [2KN_o^2 + KN_o + N_o^3]$, where m_c is the number of IAA iterations necessary to allow for convergence, and K and N_o are the

frequency grid size and the number of observed data samples. Then, the computationally efficient version of IAA method, which is named as fast segmented IAA-II(FSIAA-II), is given in [47] as $C_{FSIAA-II} = m_c[N_s^2 + (5 + 7L_s)C_{FFT}(2N_s) + (L_s + 2)C_{FFT}(K)]$, where $C_{FFT}(2N_s)$ and $C_{FFT}(K)$ denote the cost of performing FFT of lengths $2N_s$ and K , respectively, N_s is the nonoverlapping segment length ($N_s = N_o/L_s$), L_s is the number of segments, and K is the frequency grid size.

In order to provide a comparative analysis, we plotted the complexities of the ARMA lattice and transversal filters in fullband, two-subband, and four-subband cases against the complexity of the Periodogram method in Figures 5, 6, and 7, respectively. When generating a specific complexity curve for the proposed and transversal filtering methods, we assumed that the same spectrum estimation method and configuration is implemented in all subbands. Accordingly, we considered four different data lengths for the Periodogram method while allowing the filter order p to change up to 256 for ARMA(p, p), and AR(p) filters.

Since a transversal implementation does not require a Levinson–Durbin type conversion algorithm, the fast RLS transversal ARMA filtering method in subbands is computationally advantageous as compared to the proposed lattice method. The computational complexity of the proposed lattice method for ARMA(p, p) spectrum estimation compared to the Periodogram method ($N = 128$) is low as long as filter order (p) is smaller than 27

in fullband, 19 in two-subbands, and 13 in four-subbands. Similarly, the complexity for AR(p) lattice spectrum estimation compared to the Periodogram method ($N = 128$) is low as long as filter order (p) is smaller than 52 in fullband, 36 in two-subbands, and 23 in four-subbands. If longer data lengths are preferred, the low complexity threshold value of filter order for ARMA(p, p) and AR(p) implementations moves to higher values as can be observed in Figures 5, 6, and 7. We would also like to point out that it is possible to generate a family of complexity curves for each case by assuming different configurations for subband prediction filters.

We compare the computational complexities of Capon, APES, IAA, and multitaper methods with the proposed lattice ARMA(p, p) method for a data length of $N = 128$ in Figures 8, 9, and 10. When computing the computational complexity of the multitaper method, we assumed that the time-bandwidth product was $NW = 2$. Four nonoverlapping ($N_s = 32, L_s = 4$) segments, and ten iterations for convergence ($m_c = 10$) were considered for FSIAA-II. In addition, the frequency grid size for the FRLS-III type Capon and APES methods, and for the FSIAA-II method was $K = 4096$.

Accordingly, under the assumed conditions, the computational complexity of the proposed lattice method for ARMA(p, p) spectrum estimation comparing to the multitaper method is low as long as the filter order (p) is smaller than 38 in fullband, 23 in two-subbands, and 19 in four-subbands. Then, we compare the complexity

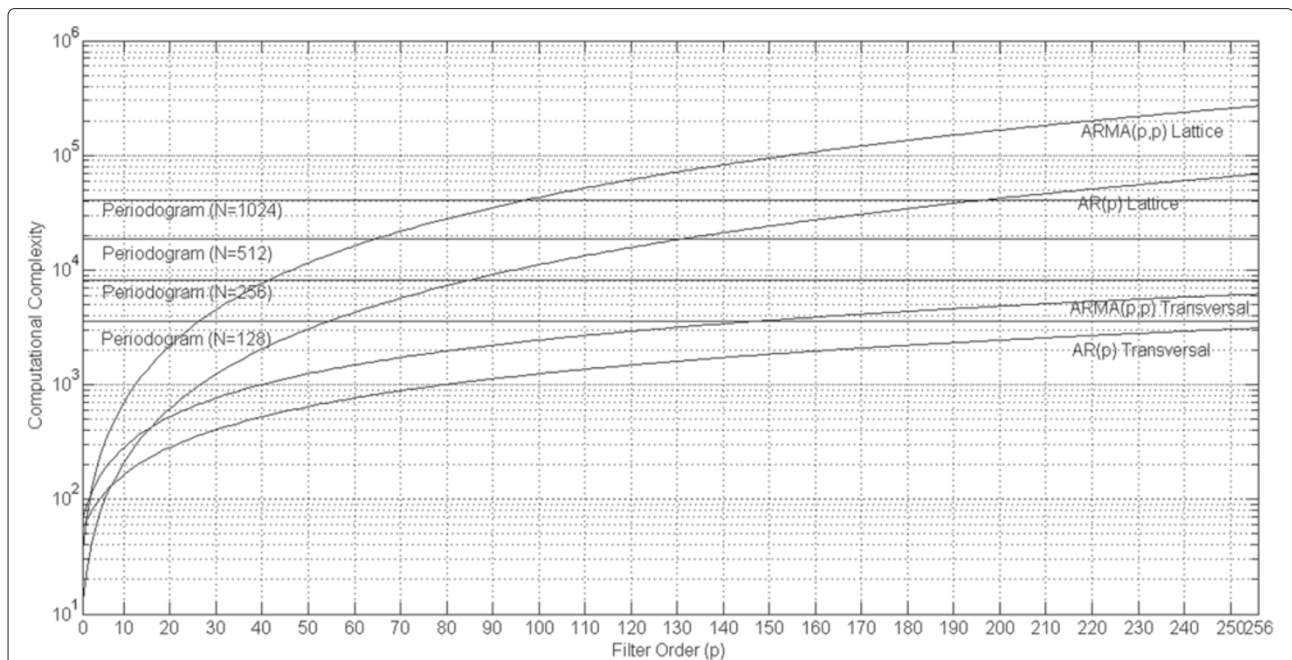
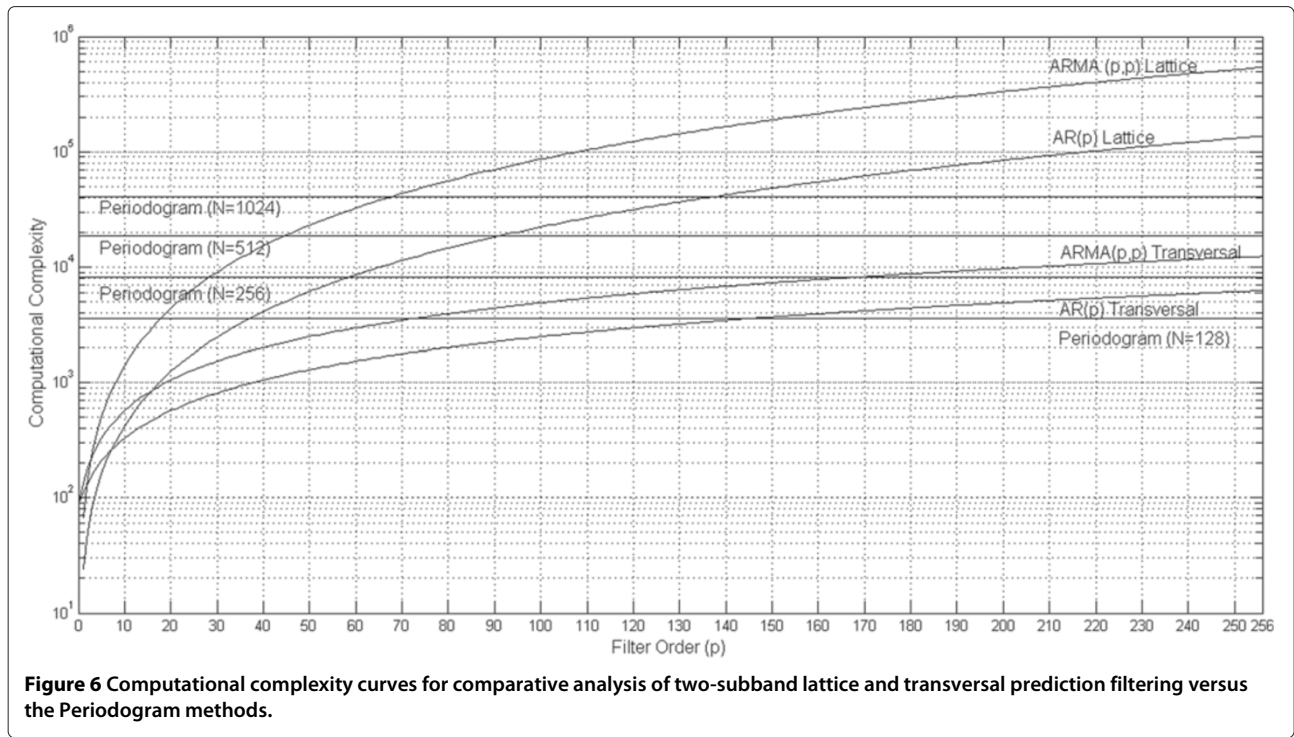
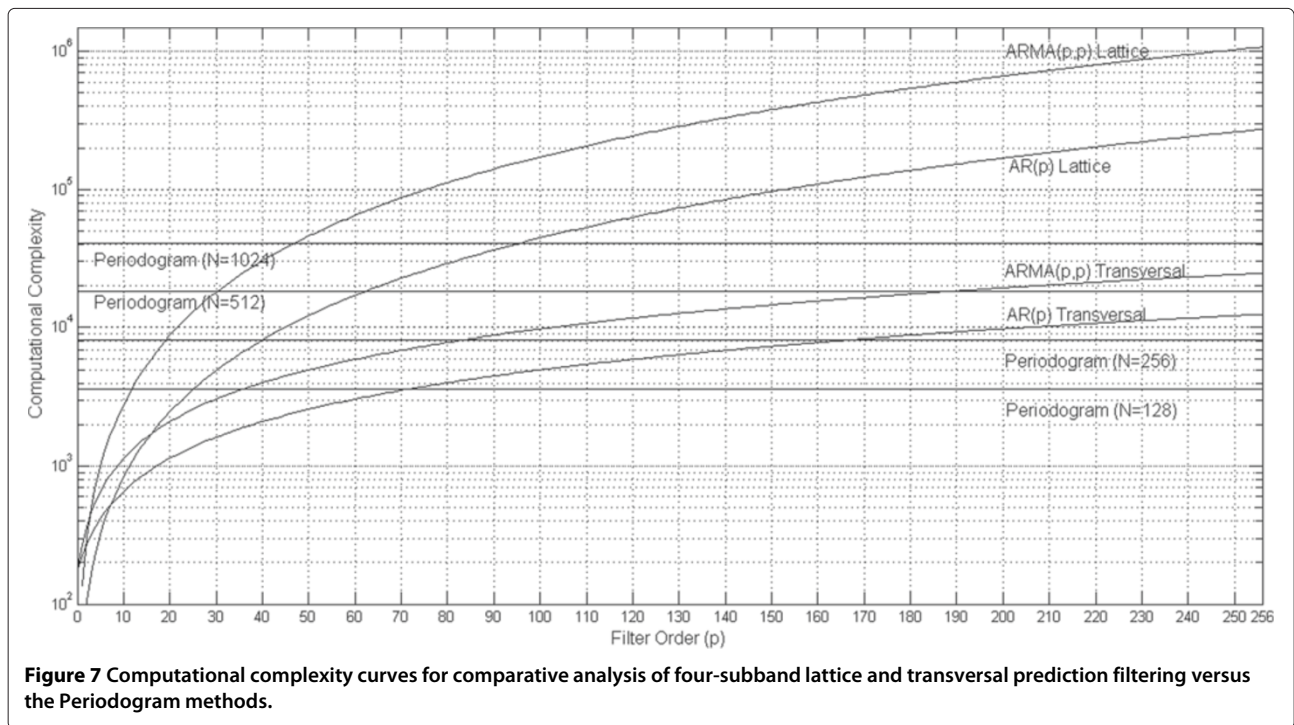


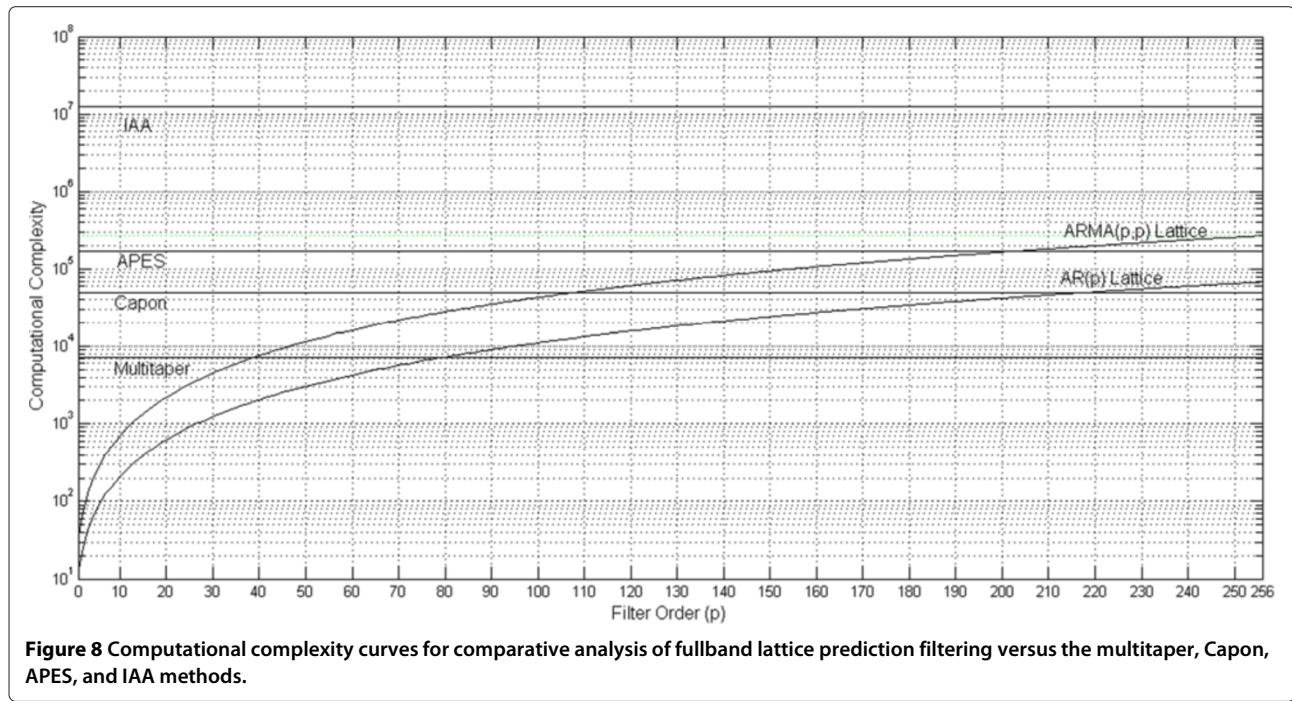
Figure 5 Computational complexity curves for comparative analysis of fullband lattice and transversal prediction filtering versus the Periodogram methods.



of proposed lattice ARMA(p,p) method with that of the Capon method, and find that its complexity is lower than the complexity of Capon method as long as the filter order (p) is smaller than 108 in fullband, 74 in two-subbands, and 55 in four-subbands. When a similar comparison

is carried out for the APES method, the computational complexity of the proposed ARMA(p,p) method is lower than the APES method as long as filter order (p) is less than 204 in fullband, 142 in two-subbands, and 100 in four-subbands. When the IAA method is considered,



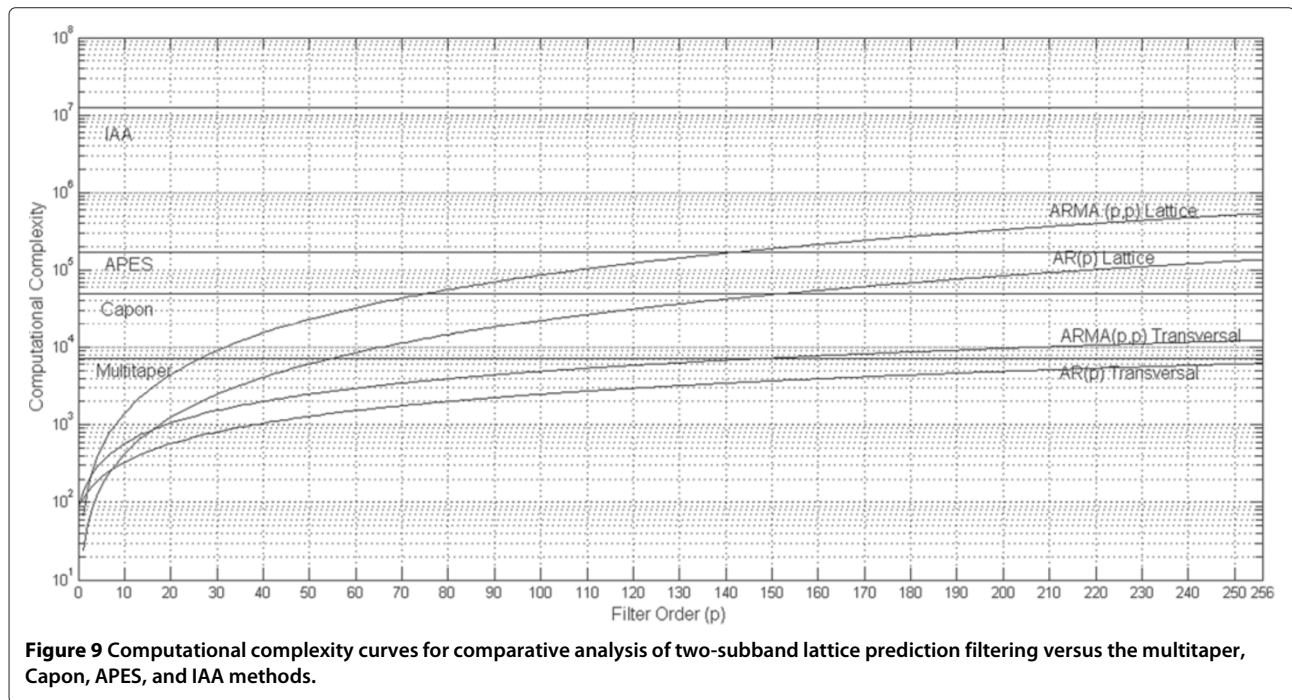


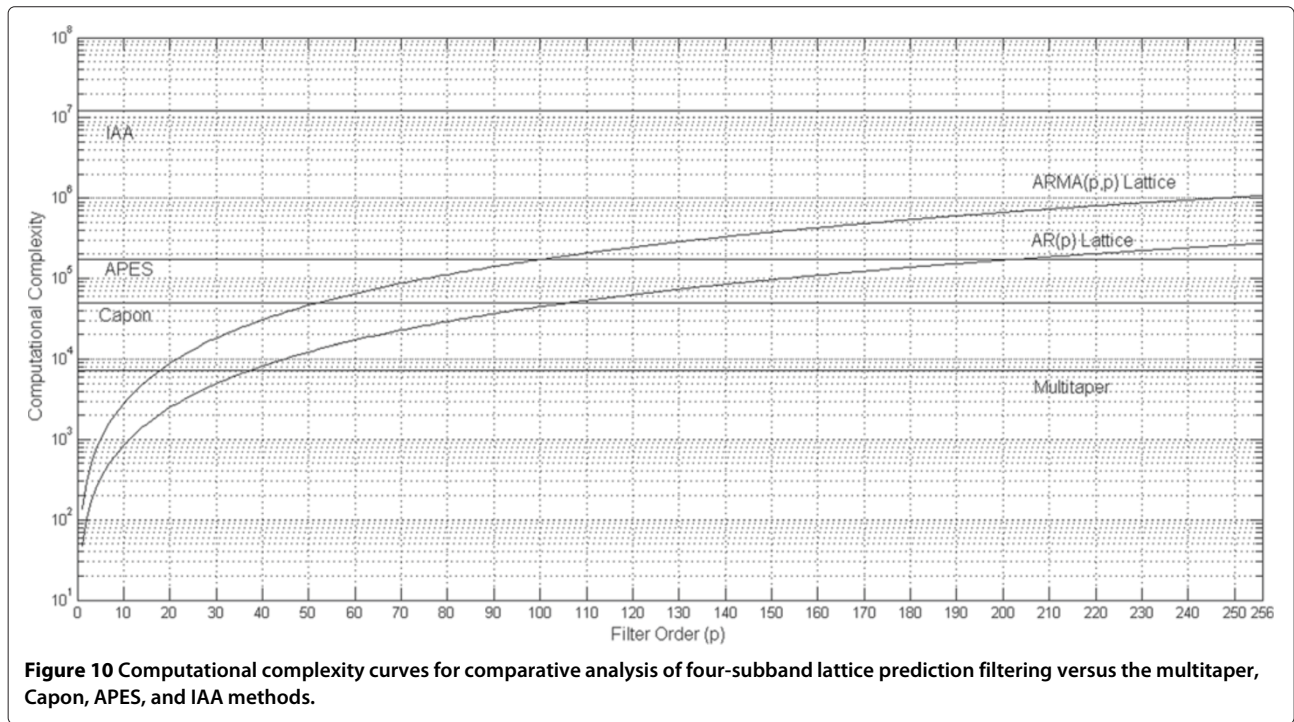
the IAA method's complexity is larger than the proposed ARMA(p,p) method as long as filter order (p) less than 3400 in fullband, 2450 in two-subbands, and 1750 in four-subbands.

6 Experimental results

We focused on ARMA(p,p) spectral estimation in simulation experiments due to its relevance in subband

implementations. Accordingly, the objectives of simulation experiments are to visually and statistically demonstrate that the proposed method has the frequency spacing improvement, whitening and SNR improvement properties, and compare its performance with the performances of nonparametric methods, viz., the Periodogram, multitaper, Capon, APES, and IAA methods. Accordingly, we present and compare the ARMA(p,p) lattice subband

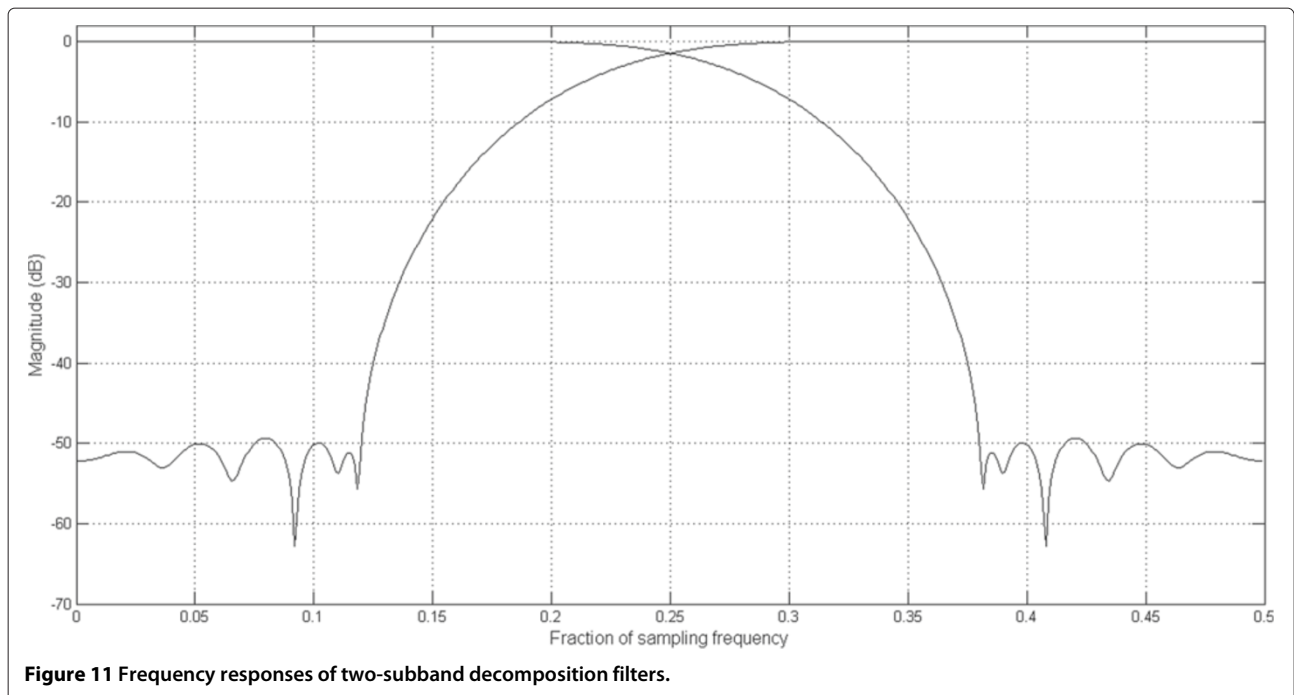




spectrum estimation results with the ARMA(p, p) lattice fullband results, and then compare the lattice four-subband results with the nonparametric results.

The forgetting factor was $\lambda = 0.995$ for stationary cases while it was smaller in nonstationary cases, $\lambda = 0.98$, so as to better track the time-varying signal. We repeated

simulation experiments one hundred times; the results of these simulations are then ensemble-averaged. In subband decomposition of input signals, we used the Kaiser window based approach in [52] for designing cosine modulated filter banks. In two- and four-subband decompositions, the filter lengths are 30 and 60, respectively,



and their frequency responses are presented in Figures 11 and 12.

We used a data length of $N = 128$, and data was zero-padded to 32 times the data length in stationary signal simulations involving the proposed lattice method, multitaper, and the Periodogram methods. In nonstationary cases, no zero-padding was utilized with any of the methods. In the comparisons with the Capon, and APES, we used the batch processing versions of these methods in [41], then we implemented the adaptive brute force versions of these methods in [46] for nonstationary signal experiments. In stationary signal cases involving the IAA method, we utilized the batch processing brute force version in [47], and subsequently in nonstationary signal cases, we made use of the adaptive brute force version in [53]. The filter lengths for the Capon and APES methods were $N_f = 63$, and we used data observation lengths of $N_o = N/2$ and $N_o = N$ in visual results and statistical analysis subsections for the IAA method, respectively. The frequency grid sizes were chosen as $K = 4096$ for the Capon, APES, and IAA methods. The sliding window size for the adaptive version of APES method was determined as $L_w = 50$. The number of IAA iterations for convergence was $m_c = 10$. We used a time-bandwidth product of $NW = 2$ for the multitaper method.

In order to determine prediction filter order in simulations, we mainly relied on our knowledge of input process order, and started with this order. Since our criteria of optimization is the minimization of forward prediction error, we increased the order of prediction filter, and monitored output forward prediction error. If the decrease in output prediction error was negligible with

the increase of filter order, we stopped increasing the filter order. Since we would not have *a priori* knowledge of process order in a practical situation, a model estimator such as ARMAset [54] can be used for this purpose. As ARMAset itself functions based on prediction error evaluations, we might not even need further prediction error evaluations. In addition to these considerations, we kept the total complexity the same in all configurations in order to provide a fair comparison of performance such that the order of fullband predictor filter was 48 while the order of predictor filters in two and four-subband implementations were 24 and 12, respectively, in all simulations.

During the simulations when the input signal is stationary, we considered two closely spaced cisoids embedded in white as well as colored noise cases. The time series for cisoids in white noise are given by

$$y(n) = A_1 e^{-j2\pi f_1 n + \phi_1} + A_2 e^{-j2\pi f_2 n + \phi_2} + u(n), \quad 1 \leq n \leq N, \quad (90)$$

where $u(n)$ is white Gaussian complex noise with uncorrelated real and imaginary parts, each with a variance σ_u^2 and zero mean, such that the SNR for the k th cisoid is defined as

$$\text{SNR}_k = 10 \log_{10} \left(\frac{|A_k|^2}{2\sigma_u^2} \right). \quad (91)$$

The time series generated by two cisoids embedded in colored noise are similarly expressed by passing the white Gaussian complex noise $u(n)$, with zero mean and unit

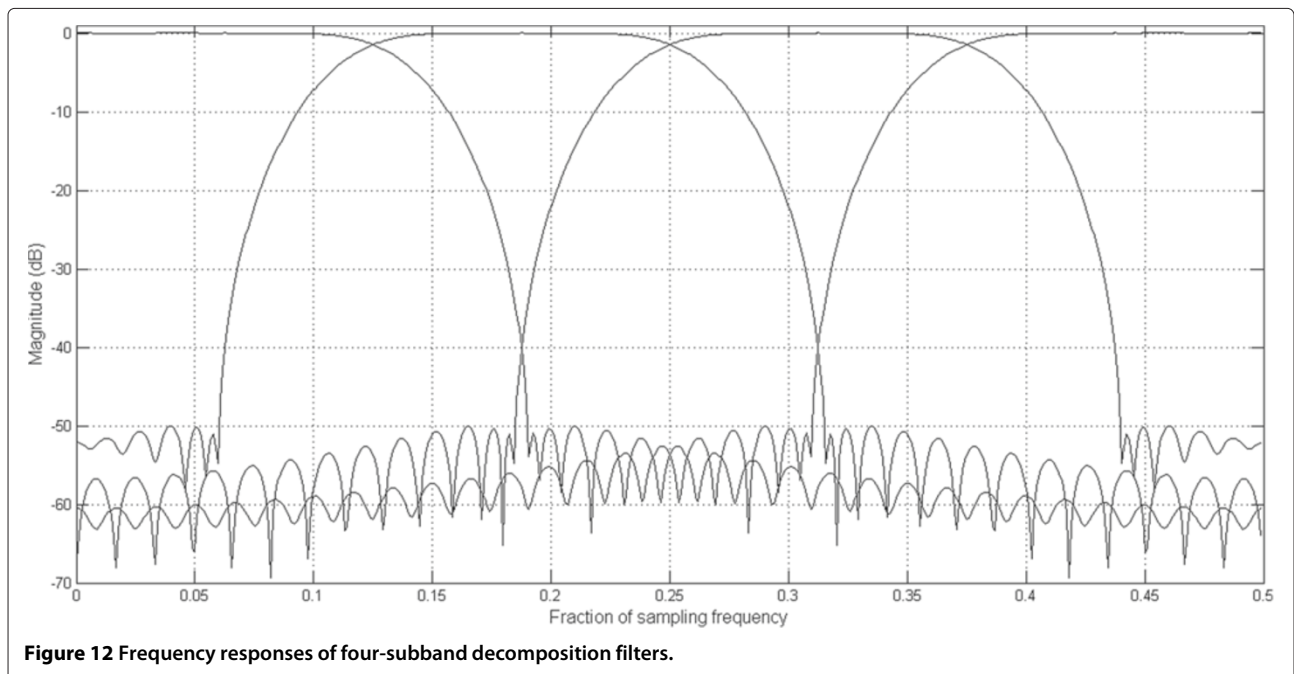


Figure 12 Frequency responses of four-subband decomposition filters.

variance, through a second-order AR coloring filter given by

$$v(n) = 0.1v(n-1) - 0.3v(n-2) + u(n), \quad 1 \leq n \leq N, \quad (92)$$

and the local SNR in this case for the k th cisoid is defined as

$$\text{SNR}_k = 10 \log_{10} \left(\frac{|A_k|^2}{V(e^{jw_k})} \right) \quad (93)$$

where $V(e^{jw_k})$ denotes the spectral density function of the AR process at the frequency of the k th cisoid. While the cisoidal frequencies were $f_1 = 0.05$ and $f_2 = 0.0656$ in the white noise case, they were $f_1 = 0.2106$ and $f_2 = 0.2262$ in the colored noise case. The initial phases ϕ_1 and ϕ_2 were zero in both cases.

In the nonstationary input signal experiments, we again considered two closely spaced cisoids embedded in white noise, but in this case, the frequencies of each individual cisoid in (90) was varied deterministically according to the following rule in the visual subsection,

$$f_k(n) = \hat{f}_k - \gamma_k \cdot (n-1), \quad k = 1, 2, \quad \text{and} \quad n = 1, \dots, N \quad (94)$$

where $\gamma_1 = \gamma_2 = 0.0001$, and $\hat{f}_1 = 0.12$ and $\hat{f}_2 = 0.1044$, so that the difference between the frequencies of cisoids is kept constant (same as the stationary signal case) during frequency sweep interval, and then according to

the random walk model of [9] in the statistical analysis subsection,

$$f_k(n) = f_k(n-1) + \gamma_k \cdot \varphi_k(n), \quad k = 1, 2, \quad \text{and} \quad (95) \\ n = 2, \dots, N$$

where $\gamma_1 = \gamma_2 = 0.001$, and $\varphi_k(n)$ is zero mean white Gaussian noise of unit variance independent of $\varphi_j(n)$, $j \neq k$, and independent of the phases ϕ_k and the measurement noise $u(n)$ for both cisoids ($k = 1, 2$). The initial frequencies of cisoids in (95) were $f_1(1) = 0.05$ and $f_2(1) = 0.0656$, respectively, and the initial phases ϕ_k in both of the nonstationary signal cases were randomly chosen in the interval $[0, 2\pi]$. The parameters, γ_k , and initial frequencies in (94) and (95) were chosen so as to keep the frequency variation inside the bandwidths of the first two- and four- subband decomposition filters during the frequency variation interval.

6.1 Visual results

6.1.1 Stationary signal case

We first consider the frequency spacing improvement simulations in which the input time series were generated using Equation (90) with no noise. The individual SNRs are infinite so that the only improvement can be due to frequency spacing. In Figure 13, we show that two-subband lattice implementation improves the frequency spacing between closely spaced cisoids, and that this improvement is even furthered by four-subband lattice

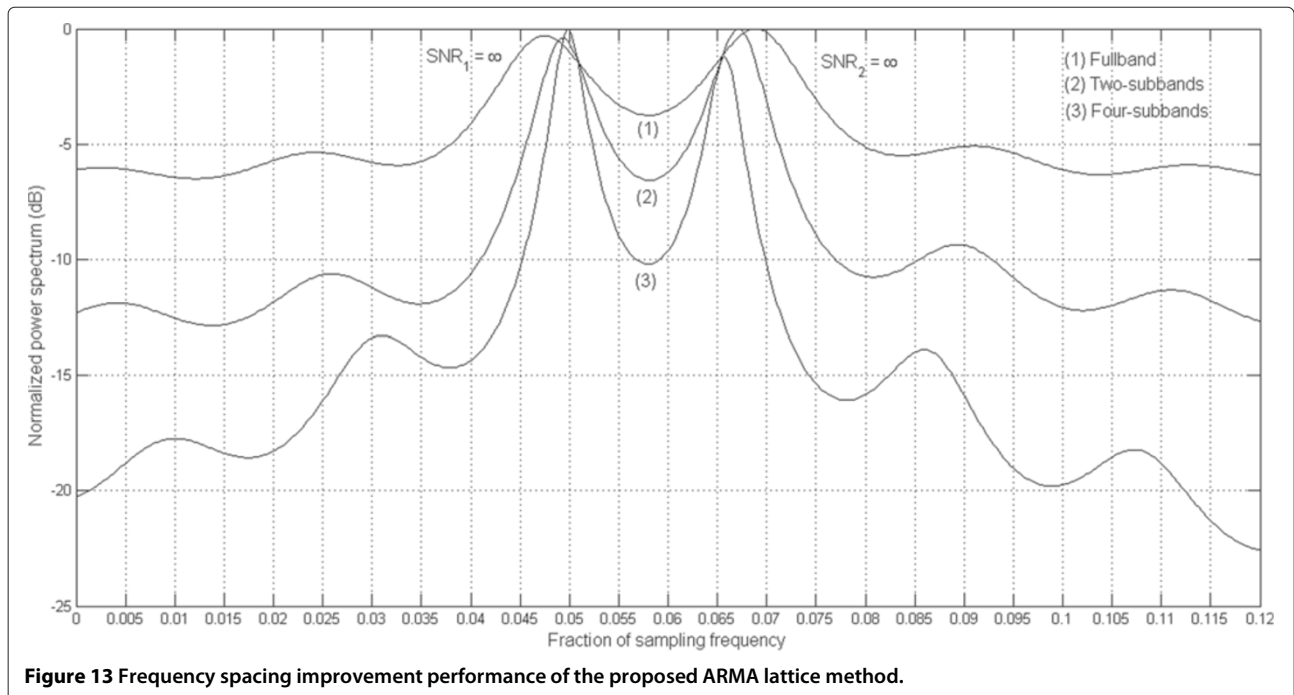


Figure 13 Frequency spacing improvement performance of the proposed ARMA lattice method.

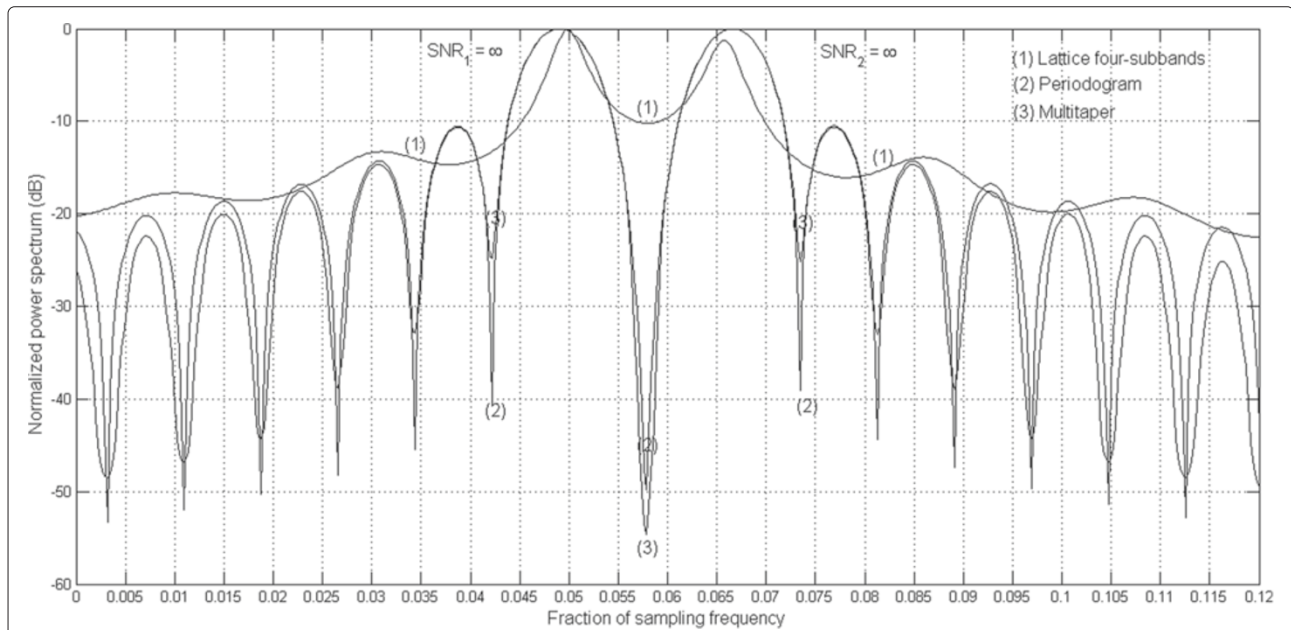


Figure 14 Comparison of frequency spacing improvement performance of the proposed method in four-subbands with the Periodogram and multitaper methods.

when compared to fullband lattice implementation. We compare the performance of the proposed method implemented in four-subbands with those of the Periodogram and multitaper methods in Figure 14, and with those of the Capon, APES, and IAA methods in Figure 15.

In the SNR improvement simulations, we again used the input time series generated by (90) with relatively low SNRs, $SNR_1 = SNR_2 = -3$ dB, so that the SNR improvement can be better displayed. Figure 16 illustrates that two-subband lattice implementation is able to resolve cisoids better comparing to fullband lattice implementation, and that four-subband implementation can resolve

them much better due to SNR improvement effect of subband filtering. We show the performance of the proposed method in four-subbands together with the Periodogram and multitaper methods in Figure 17, and then with the Capon, APES, and IAA methods in Figure 18.

The next property to be considered is the whitening property of subband filtering. In this experiment, we once again generated the input time series using Equation (90), however the white noise $u(n)$ was replaced with the colored noise $v(n)$ in Equation (92). The local SNRs for the cisoids are $SNR_1 = SNR_2 = 0$ dB in this case. In Figure 19, we compare the estimated spectrums based on

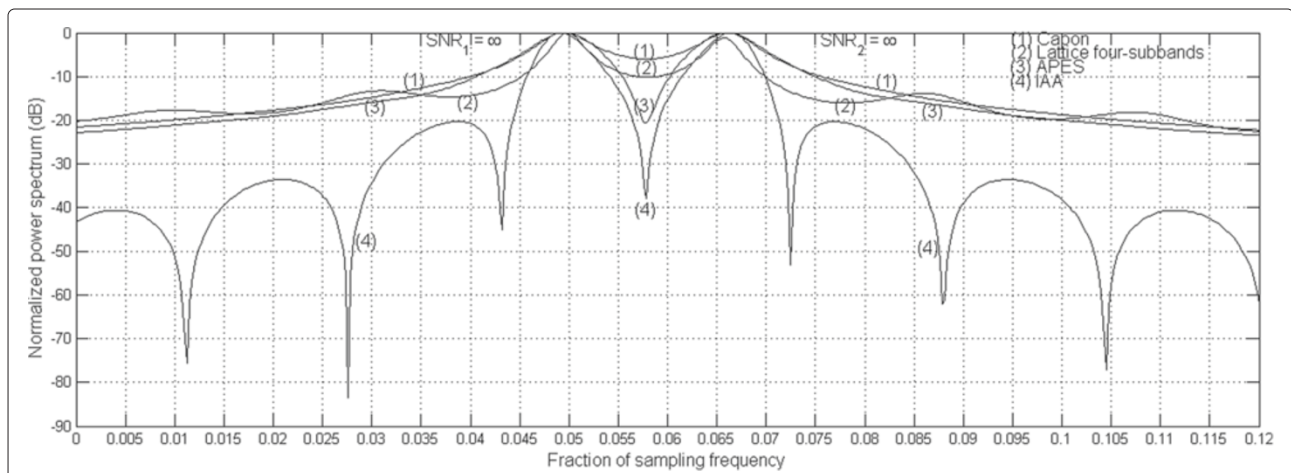
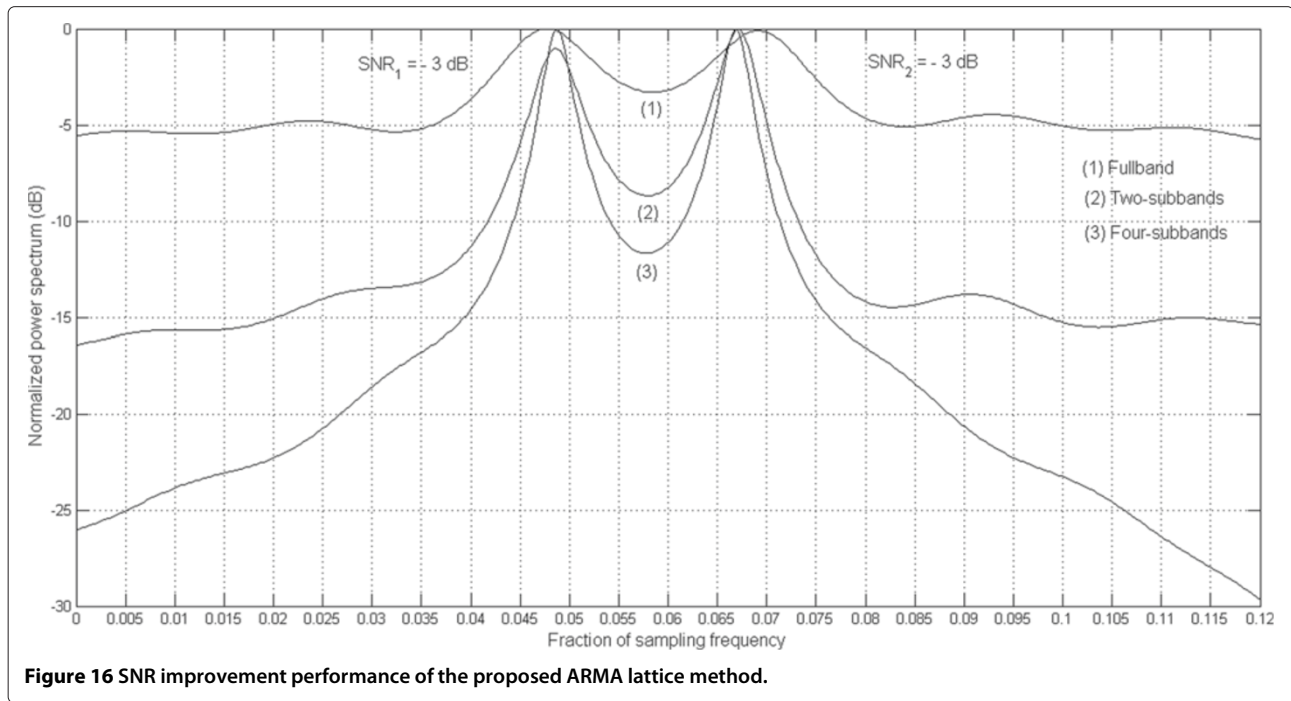


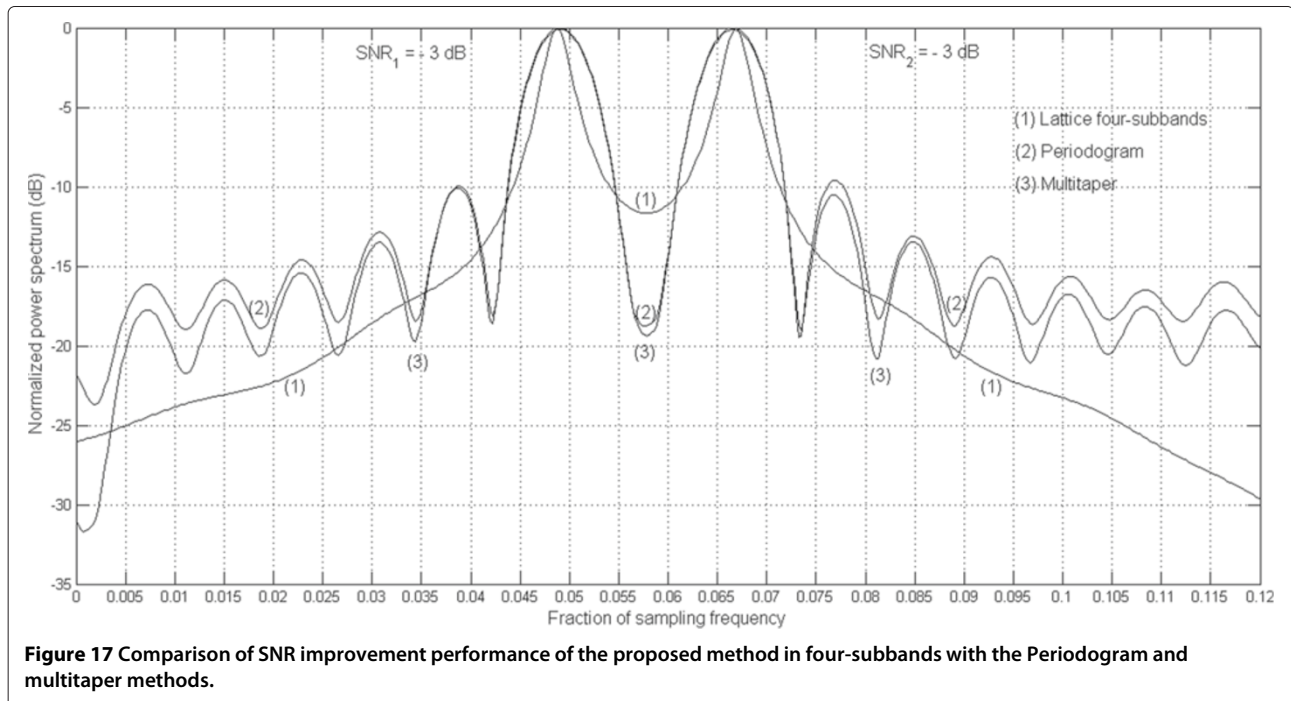
Figure 15 Comparison of frequency spacing improvement performance of the proposed method in four-subbands with the Capon, APES, and IAA methods.

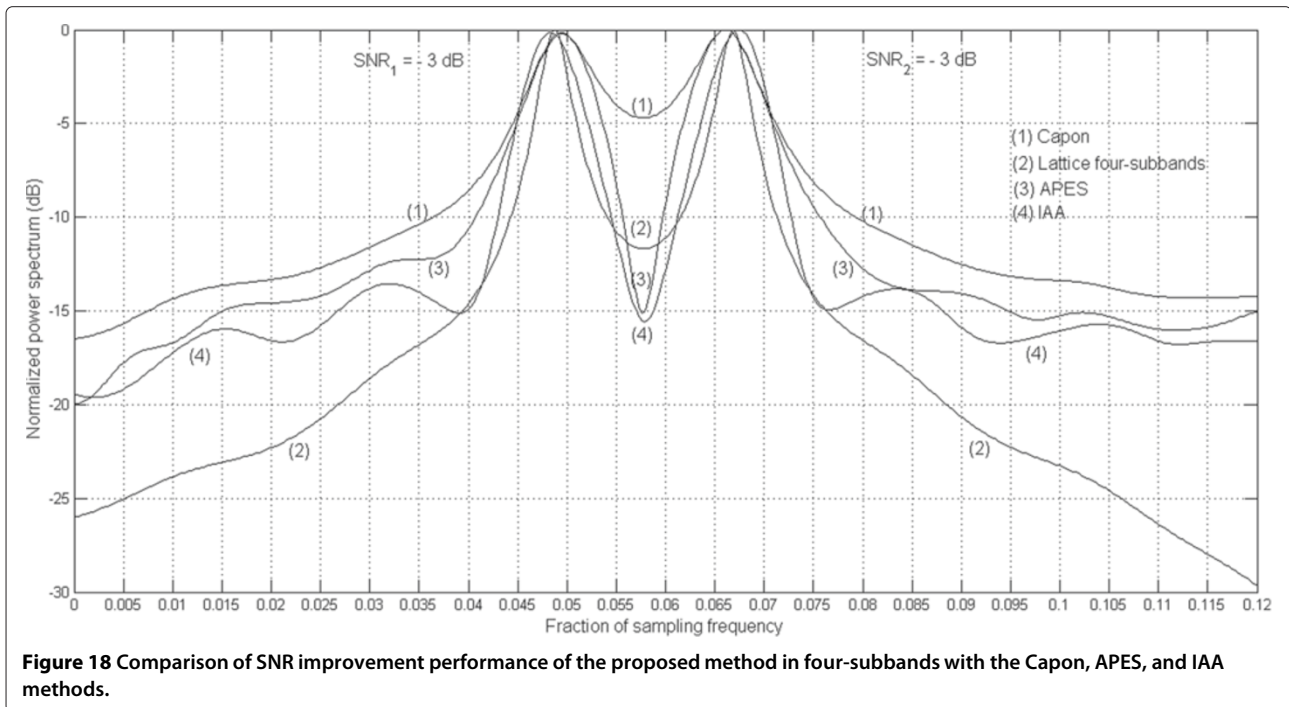


fullband, two-, and four-subbands ARMA lattice filtering. Evidently, the cisoids are better resolved in the two- and four-subband spectrums when compared to the fullband spectrum. Also, they are more clearly resolvable in four-subbands than in two-subbands. Figure 20 demonstrates the comparison of the proposed method in four-subbands with the Periodogram method and multitaper methods,

and Figure 21 does the comparison with the Capon, APES, and IAA methods.

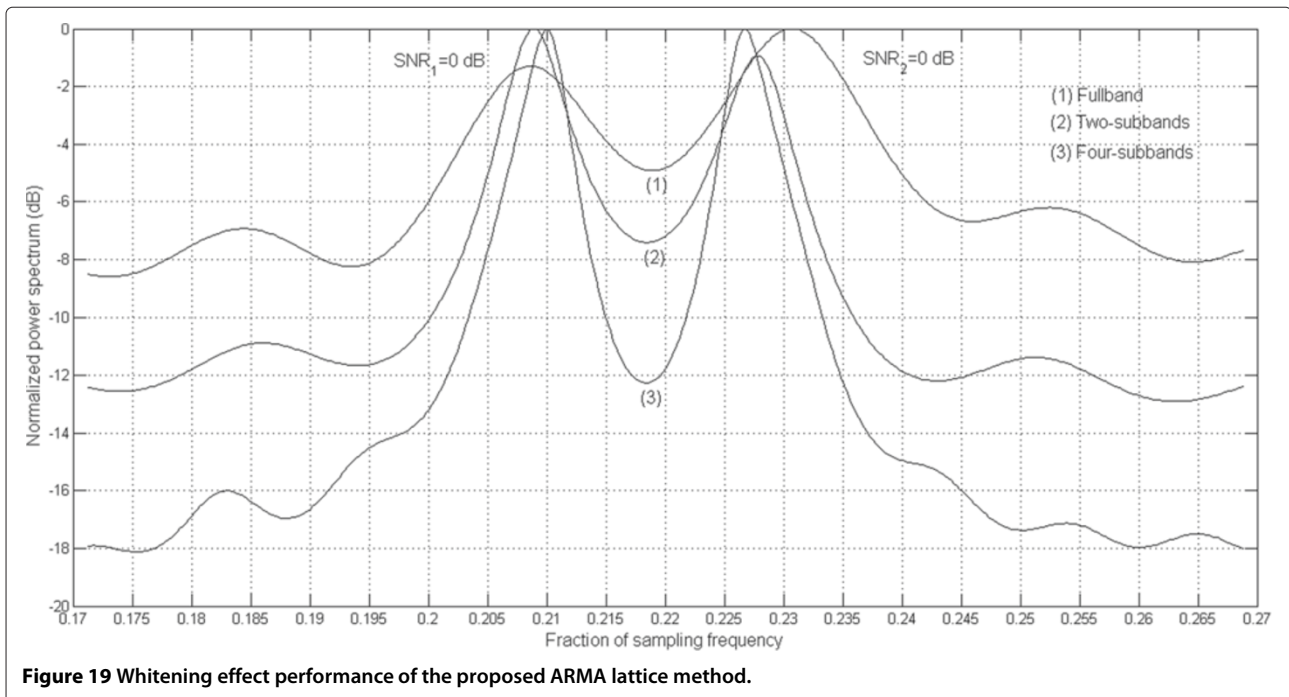
In Figures 14, 17, and 20, we see that nulls between cisoids are deeper in the Periodogram and multitaper spectrums than those of the four-subband lattice method; the proposed lattice method in four-subband results in smoother spectrums with narrow mainlobes and low

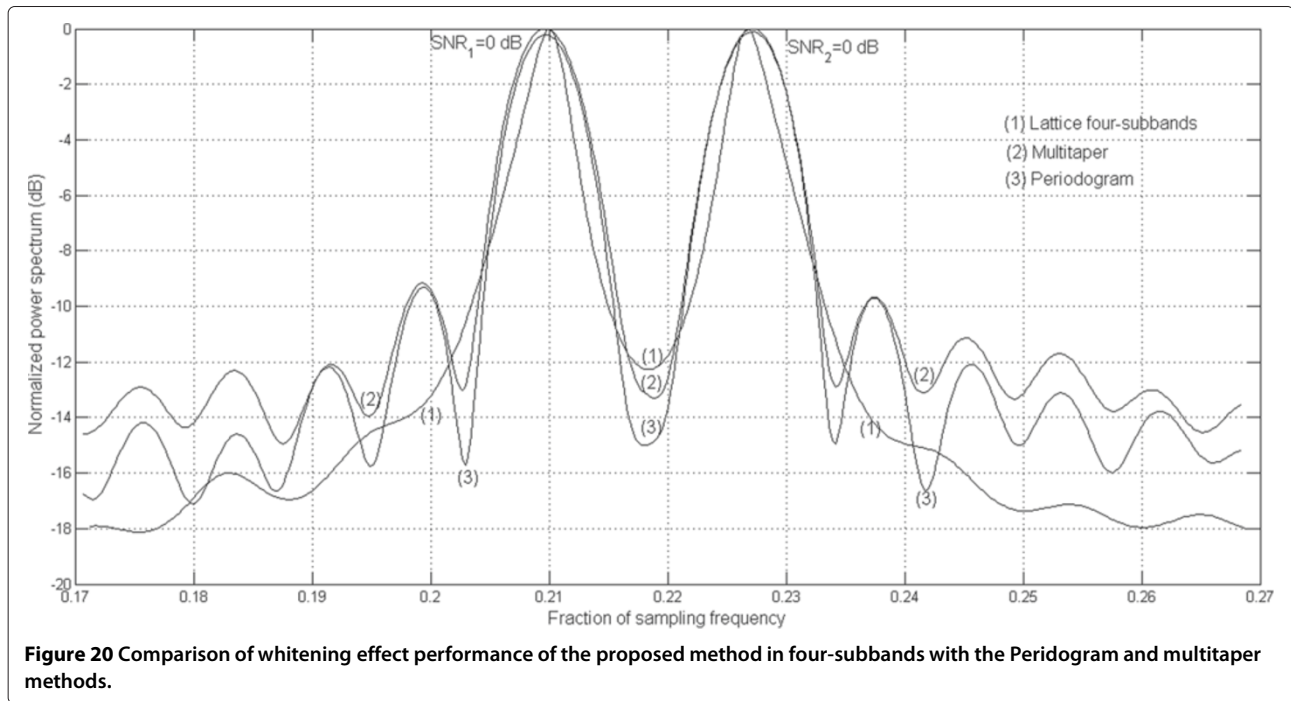




sidelobes while the Periodogram and multitaper spectrums have broader mainlobes and higher sidelobes. We then compare the lattice four-subband spectrums with those of the Capon, APES, and IAA in Figures 15, 18, and 21. It can be seen in these figures that the Capon spectrums have the highest null between cisoids among all of the methods, but they are smooth with relatively broader

mainlobes and higher sidelobes comparing to the lattice four-subband spectrums. The APES spectrums have deeper nulls between cisoids than the lattice four-subband spectrums, and they are also smooth with broader mainlobes and higher sidelobes than the lattice four-subband spectrums. The IAA spectrums have the lowest nulls, but their mainlobes are broader and their sidelobes are



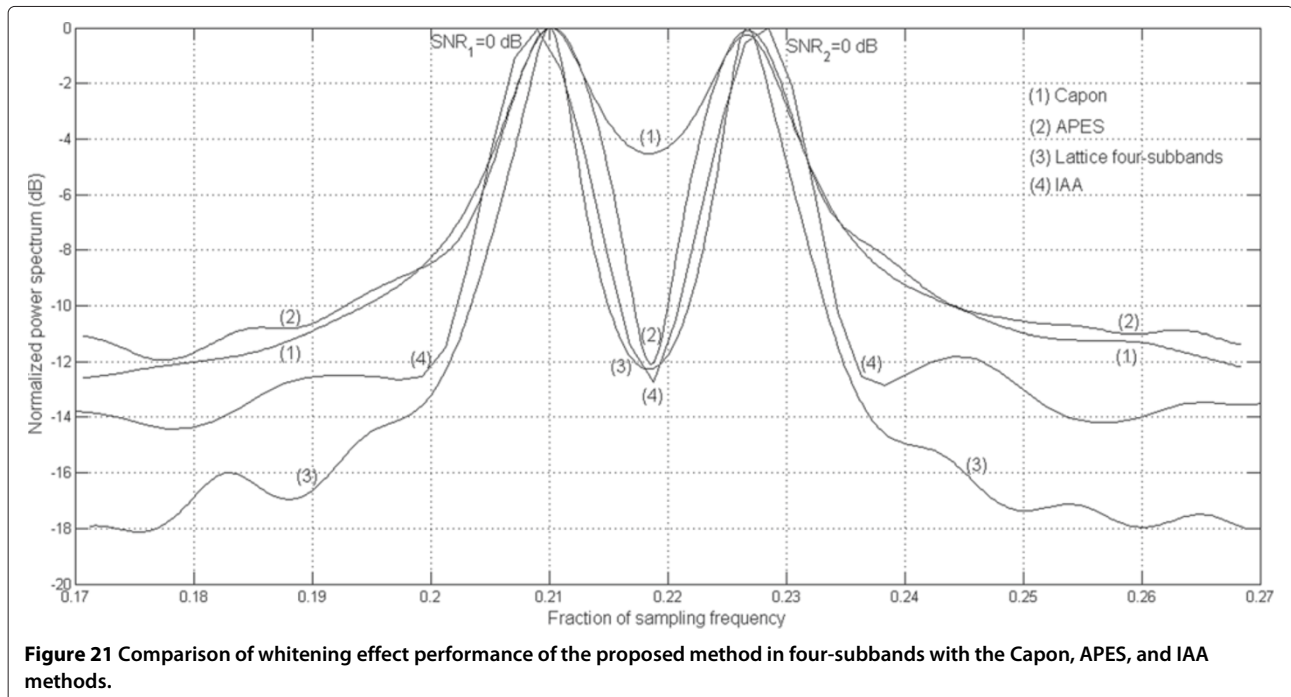


higher in SNR improvement and whitening experiments comparing to the proposed method using four-subbands.

6.1.2 Nonstationary signal case

We again consider the input time series that were generated using Equation (90), but cisoidal frequencies

are varied deterministically according to Equation (94). $SNR_1 = SNR_2 = 15$ dB was assumed. In Figure 22, we plotted one of the frequencies estimated by the proposed lattice method in fullband, two-subbands, and four-subbands against the true frequency trajectory as a function of number of iterations. It can be observed that the lattice four-subband estimates are closer to the



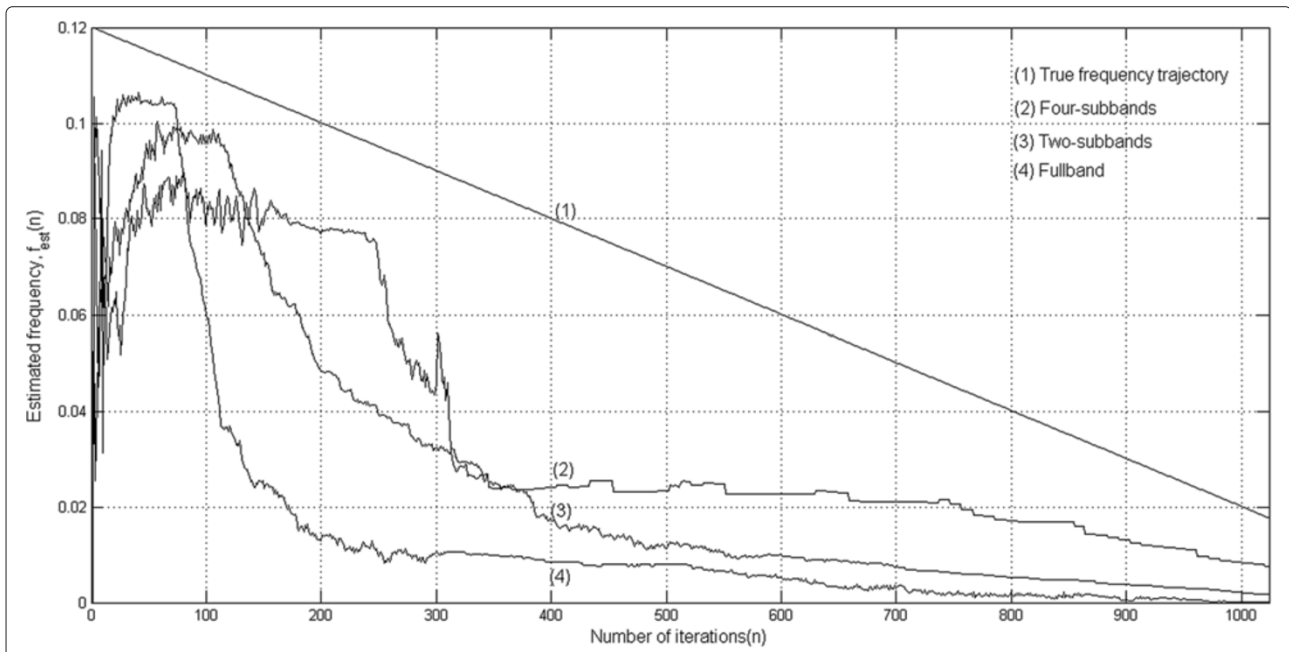


Figure 22 Estimation of time-varying frequencies of two cisoids embedded in white noise with the proposed ARMA lattice method.

true frequency values and also converges earlier than the lattice two-subband and fullband estimates. Figures 23, 24, and 25 compare the lattice four-subband frequency estimates with those of the Capon, APES, and IAA estimates, respectively. Note that the Capon method converges earlier than the lattice four-subband method, and the converge behavior of the APES method is similar to

the proposed method using four-subbands, however its values at the end of converge duration are farther away from the true frequency values comparing to those of the lattice method. Figure 25 illustrates that the IAA method converges earlier than the proposed method using four-subbands after a rather fluctuating convergence behavior.

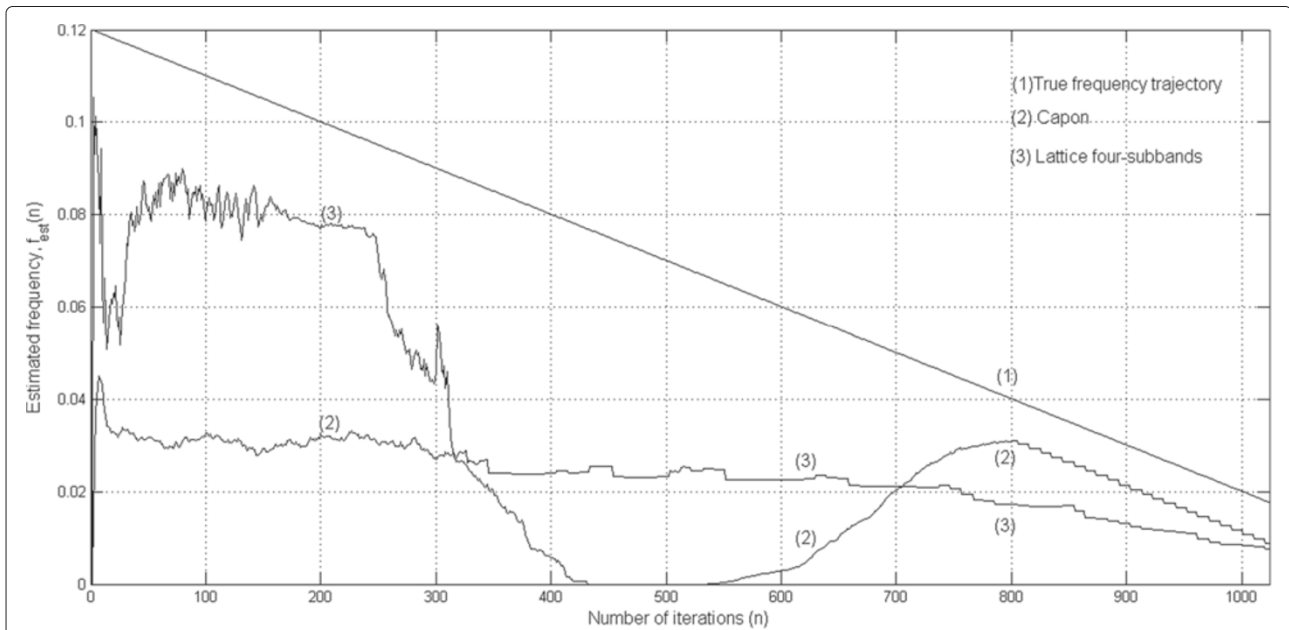


Figure 23 Comparison of the proposed method in four-subbands with the Capon method by means of estimated time-varying frequencies of two cisoids embedded in white noise.

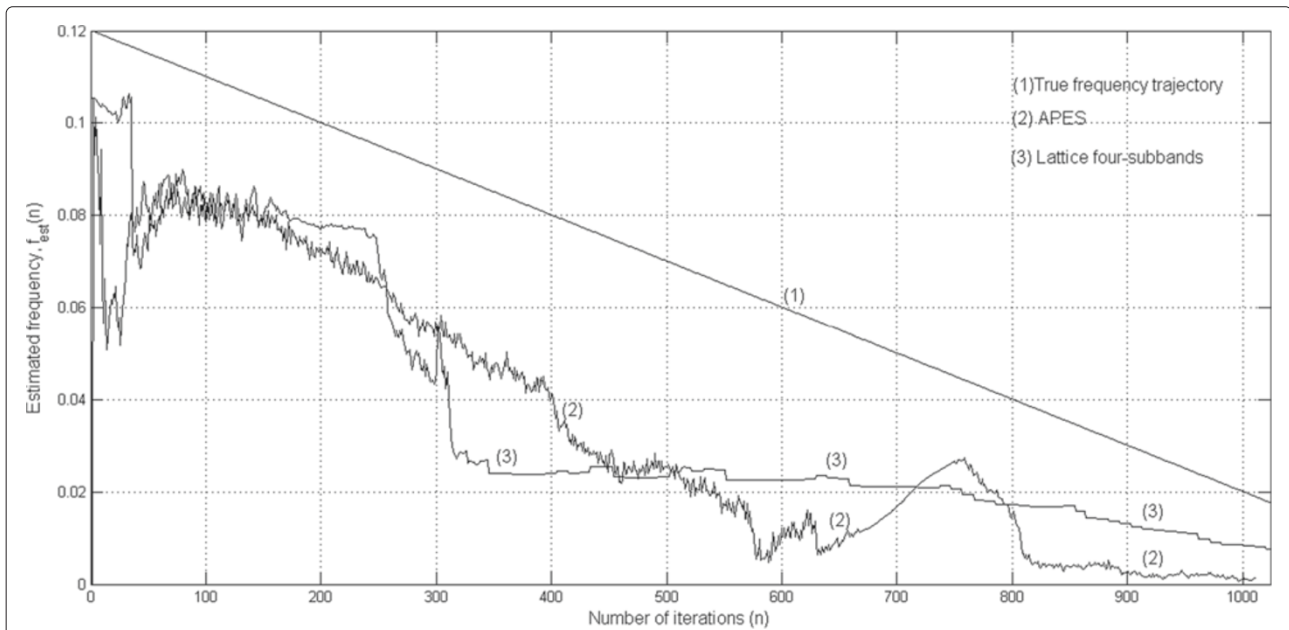


Figure 24 Comparison of the proposed method in four-subbands with the APES method by means of estimated time-varying frequencies of two cisoids embedded in white noise.

6.2 Statistical analysis

We also consider the statistical analysis section under stationary and nonstationary signal cases. In stationary signal case, we investigate the parameter estimation performance of the proposed method by performing variance and mean bias simulations for SNRs ranging from -20 to 20 dB with 1 dB increments. We assumed that the

number of cisoids are known in these simulations, and then compared the variance performance of the proposed method against the Cramer-Rao bound (CRB) in [55]. In nonstationary signal case, we provide the parameter estimation performance by means of variance and mean bias versus number of iterations, and also compare against the CRB as a function of number of iterations.

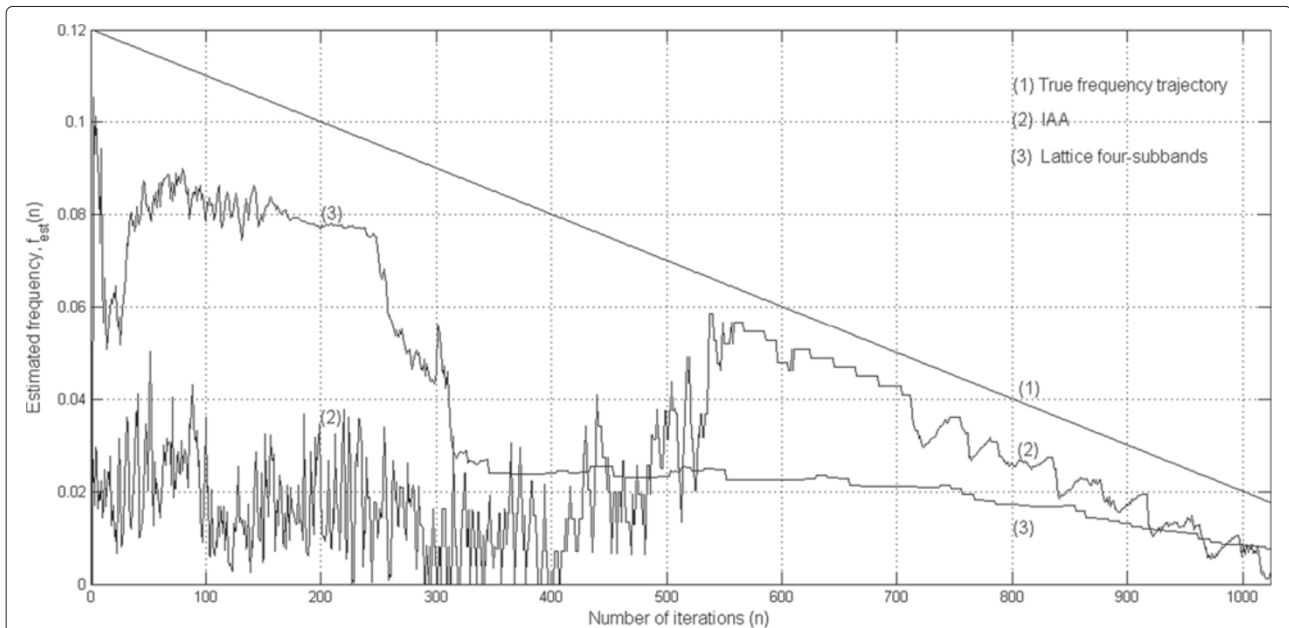


Figure 25 Comparison of the proposed method in four-subbands with the IAA method by means of estimated time-varying frequencies of two cisoids embedded in white noise.

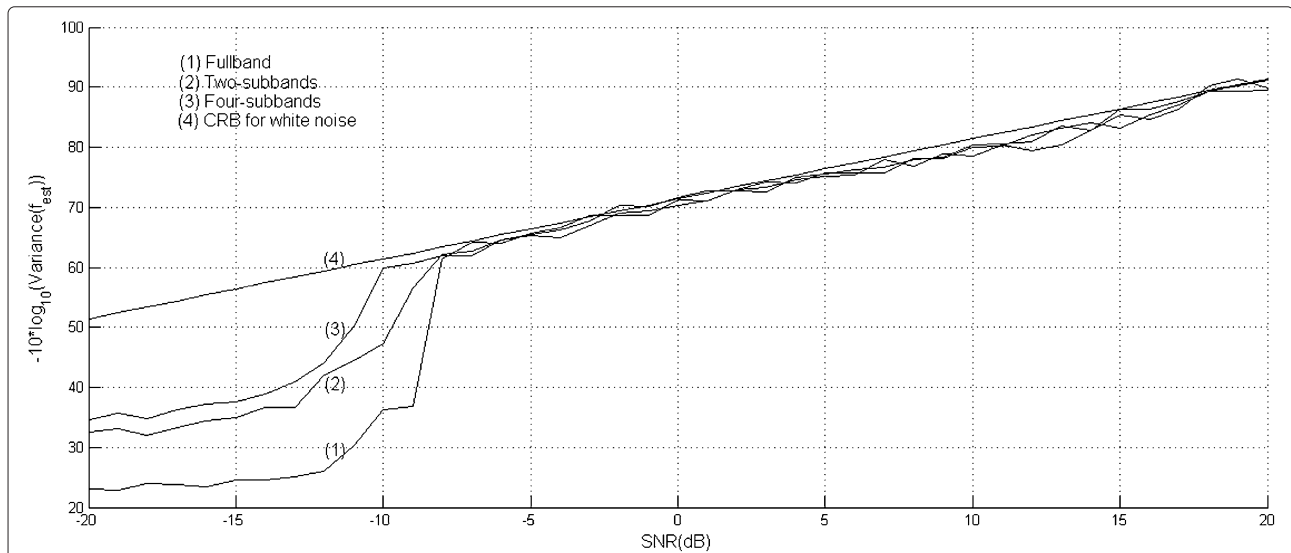


Figure 26 Variance versus SNR plots for estimating the frequencies of two cisoids embedded in white noise with the proposed ARMA lattice method.

6.2.1 Stationary signal case

In the first experiment, the input time series were generated by Equation (90) for two closely spaced cisoids in white noise. We present the variance curves of the proposed lattice method in Figure 26, and then compare the lattice four-subband variance curve with the variance curves generated using the Periodogram and multitaper methods in Figure 27, and the Capon, APES, and IAA methods in Figure 28, respectively. It can be observed

in Figure 26 that subband filtering improvement effects are more pronounced in the low-SNR region ($-20 \text{ dB} \leq \text{SNR} \leq -10 \text{ dB}$), and the threshold SNR, below which the estimation accuracy degrades rapidly, is about -9 dB . In the higher-SNR region ($-10 \text{ dB} < \text{SNR} < 10 \text{ dB}$), the subband estimation variance curves cross over the CRB while saturating earlier (at approximately $\text{SNR} = 7 \text{ dB}$ and $\text{SNR} = 9 \text{ dB}$ in two- and four-subbands, respectively) than the fullband estimation variance curve due

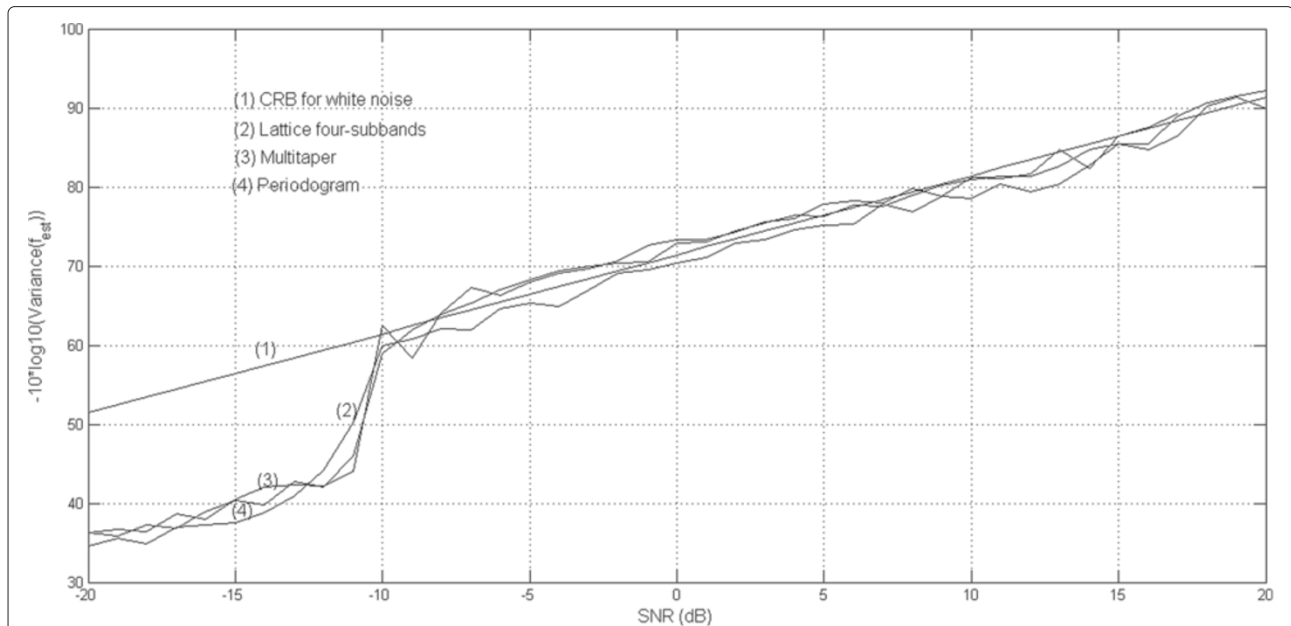
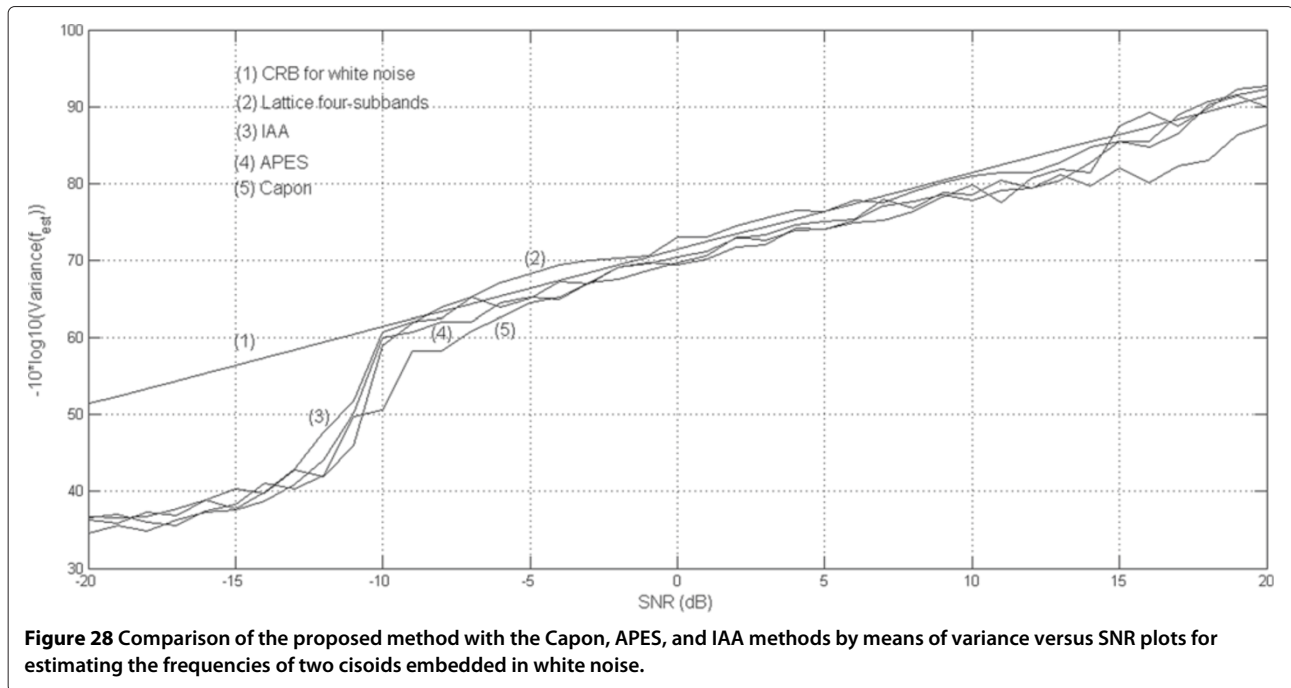
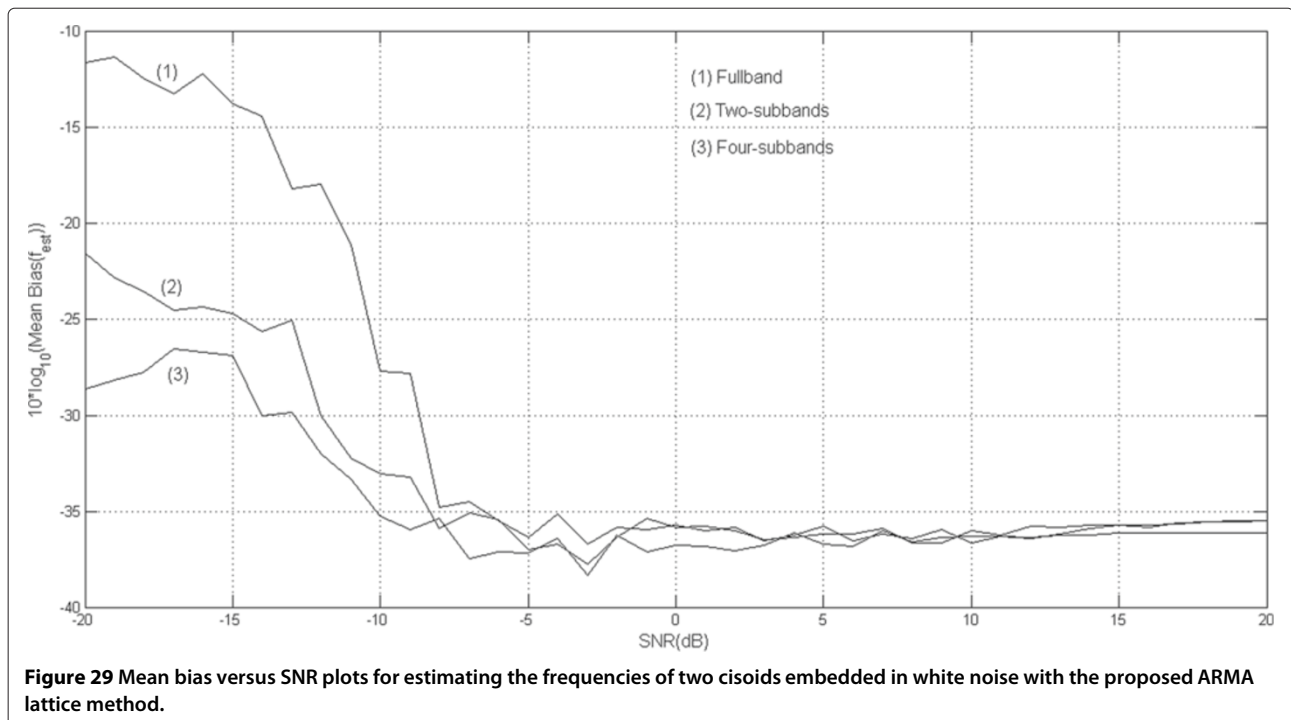


Figure 27 Comparison of the proposed method with the Periodogram and multitaper methods by means of variance versus SNR plots for estimating the frequencies of two cisoids embedded in white noise.



to the shorter data lengths in subbands and the quantization effects induced by the sampling of the frequency variable. The saturation point can be moved to higher SNRs and accordingly the estimation accuracy can be improved by using longer data and more zeropadding at the cost of more computation time. When we compare the lattice four-subband variance curve with those

of the Periodogram and multitaper methods in Figure 27, we can see that the lattice four-subband variance values are very close to those of the Periodogram and multitaper methods throughout all SNR regions. In Figure 28, the variance curve of the proposed method using four-subbands is plotted against those of the Capon, APES, and IAA methods. It can be observed that the variance



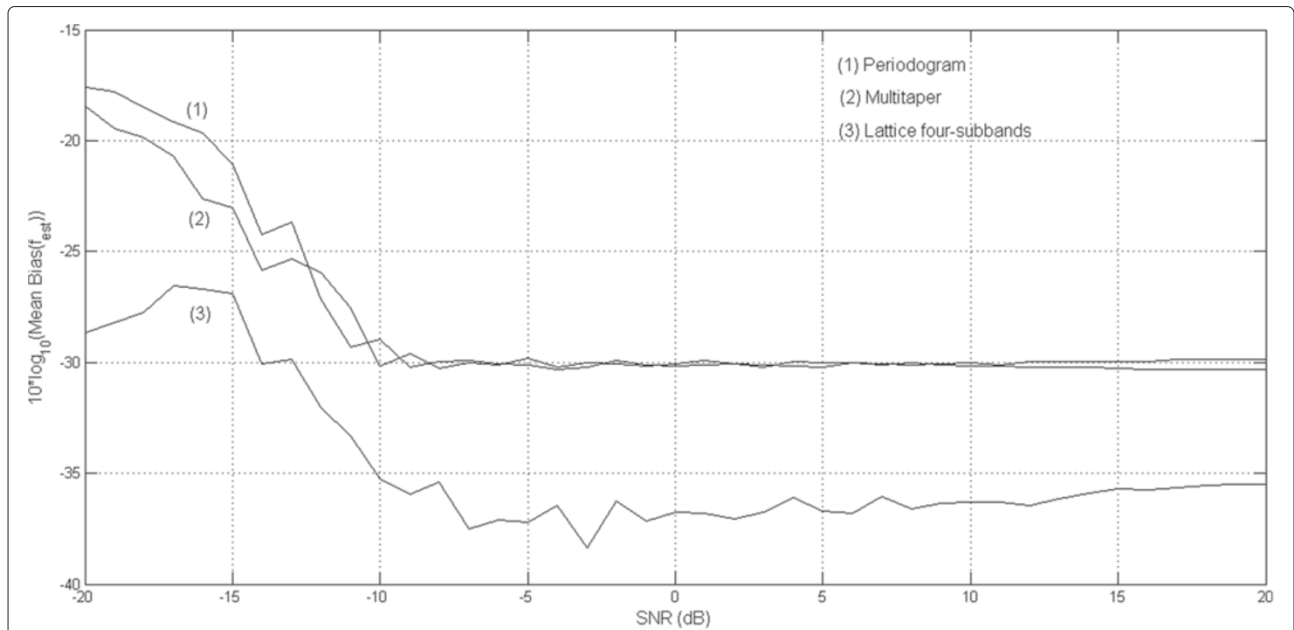


Figure 30 Comparison of the proposed method with the Periodogram and multitaper methods by means of mean bias versus SNR plots for estimating the frequencies of two cisoids embedded in white noise.

curve of the proposed method using four-subbands is higher than those of the Capon, APES, and IAA in the medium SNR region ($-10 \text{ dB} < \text{SNR} < 5 \text{ dB}$), and close to those of the Capon, APES, and IAA in the rest of SNR range.

In Figure 29, we present mean bias curves for fullband as well as subband implementations for two cisoids

in white noise. Similar to the variance curves, the mean bias values of the proposed method in fullband and two-subbands are relatively higher than those of the proposed method in four-subbands in the low-SNR region, and the mean bias values of the proposed method in two-subbands are also higher than those of the proposed method in four-subbands in the same SNR region, and

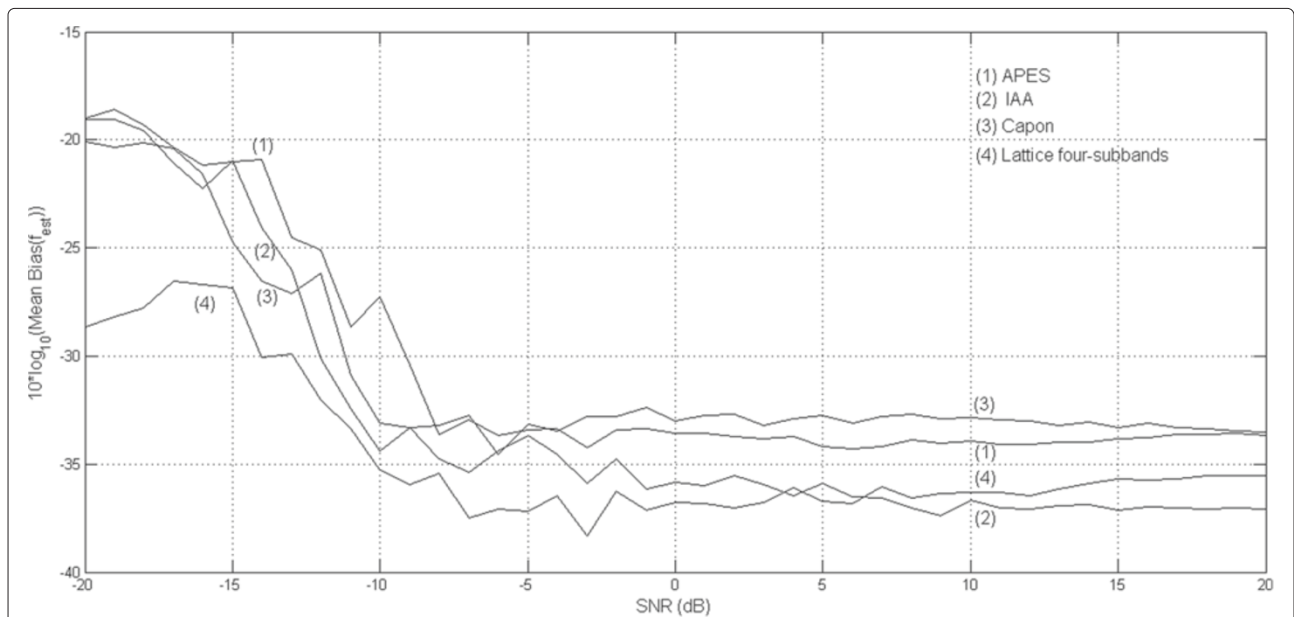
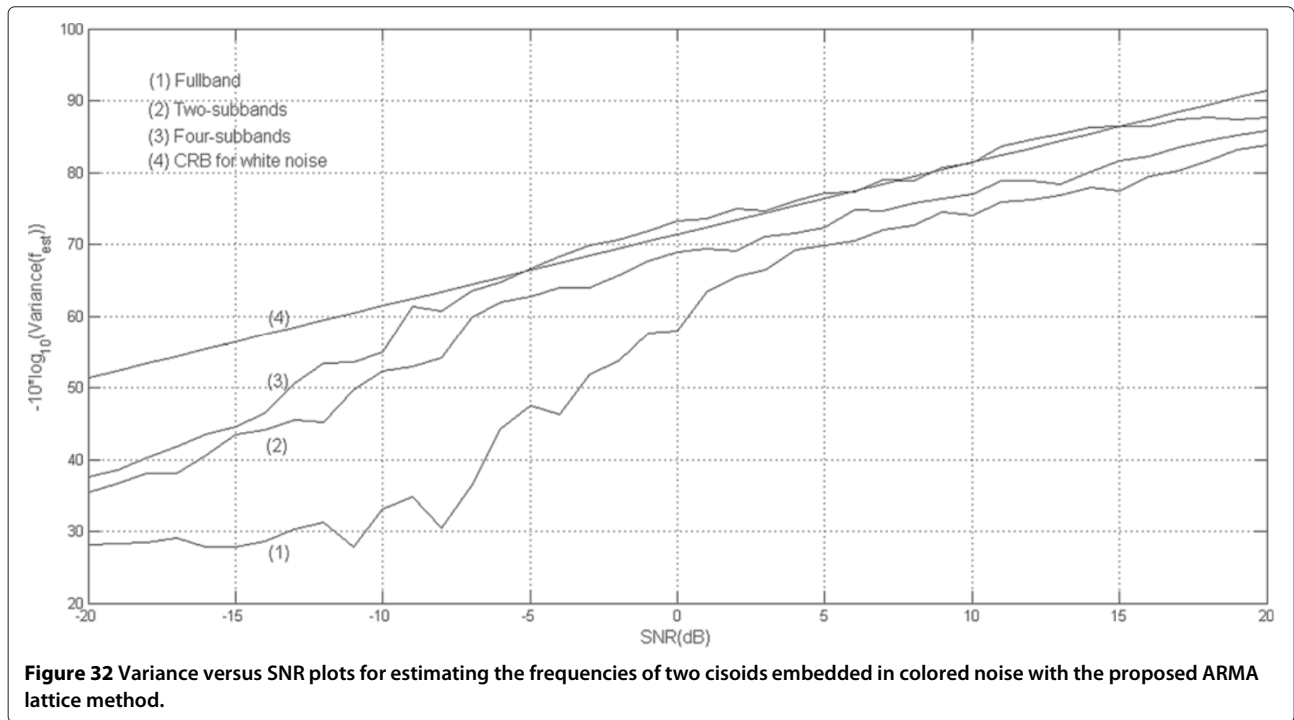
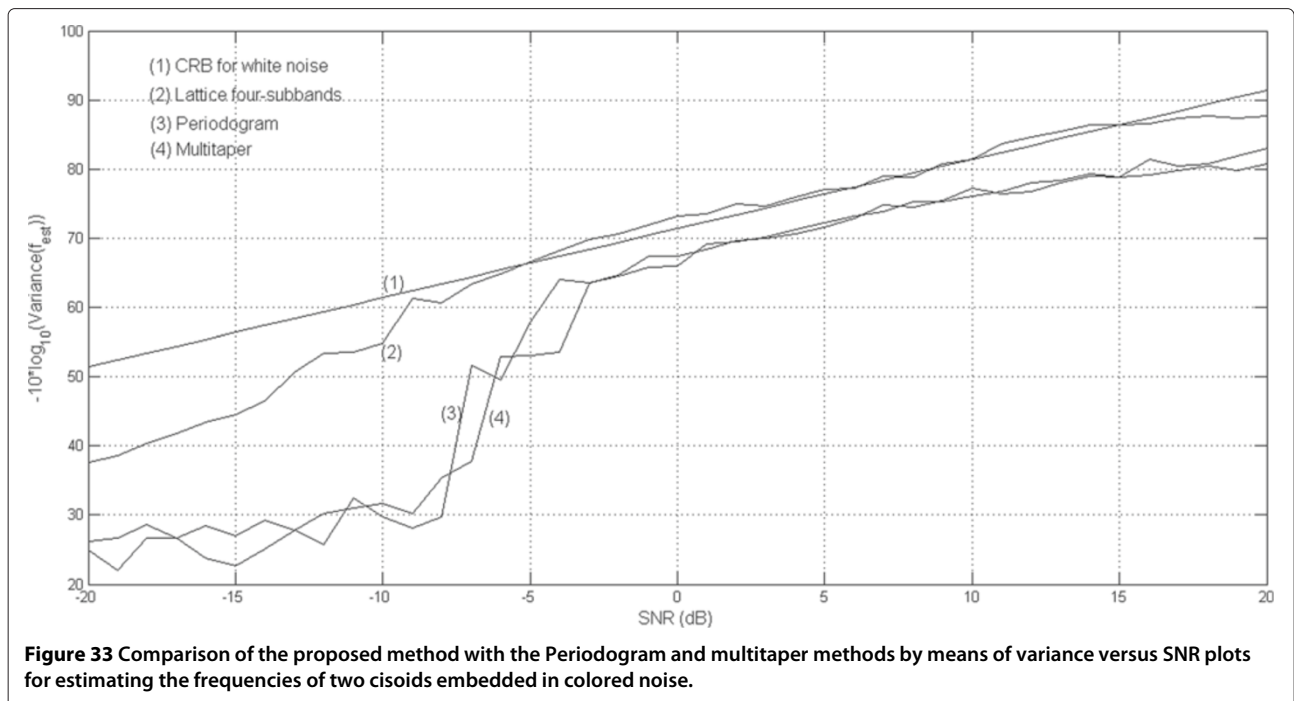


Figure 31 Comparison of the proposed method with the Capon, APES, and IAA methods by means of mean bias versus SNR plots for estimating the frequencies of two cisoids embedded in white noise.



in the higher-SNR region ($\text{SNR} > -9$), the proposed method in fullband, two-subbands, and four-subbands display similar mean bias behaviors. In Figure 30, it can be seen that the Periodogram and multitaper have higher mean bias values comparing to the lattice four-subband method in the full SNR range, and in Figure 31, it can

be observed that the lattice method in four-subbands performs better than the Capon, APES methods, and IAA methods in the SNR region ($-20 \text{ dB} \leq \text{SNR} \leq 5 \text{ dB}$) while it is better than the Capon and APES methods and slightly worse than the IAA method in the rest of SNR region.



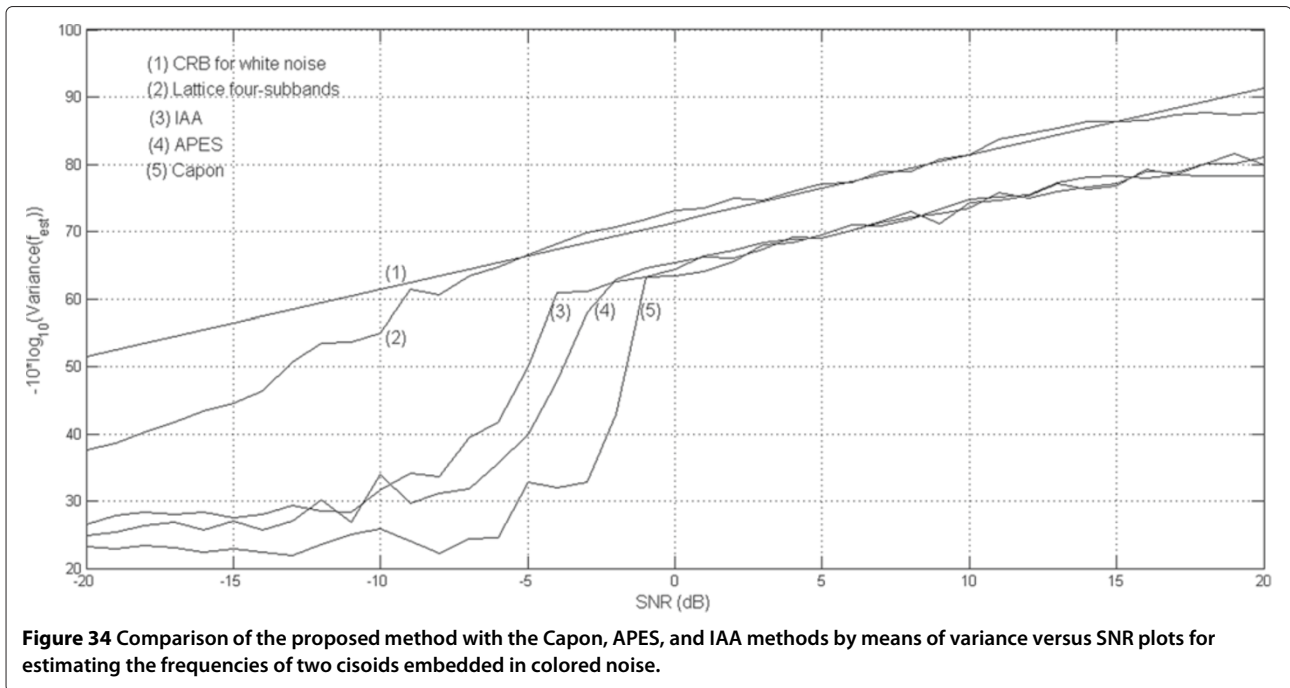
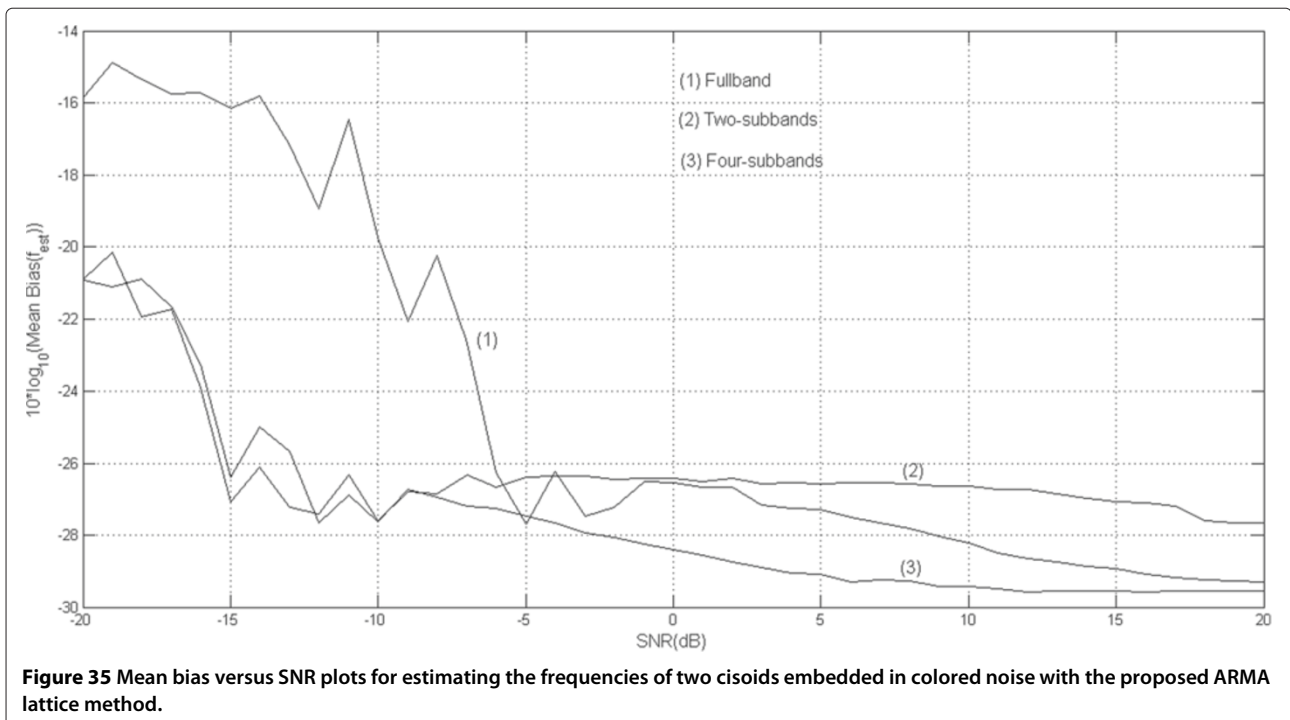


Figure 32 demonstrates variance curves of the proposed lattice method when the input time series were generated by Equation (90) with the colored noise $v(n)$ in Equation (92) replaced the white noise $u(n)$. In Figure 32, it can be observed that the threshold local SNR moved to approximately SNR = 2 dB in fullband, SNR = -7 dB in two-subbands, and -9 dB in four-subbands. Again, subband

filtering effects are more pronounced in the low-SNR region while they relatively improve the estimation performance in the higher-SNR region. Note that only four-subband variance curve crosses the CRB in the colored additive noise simulation. We compare the variance performance results of the proposed lattice method using four-subbands with those of the Periodogram and



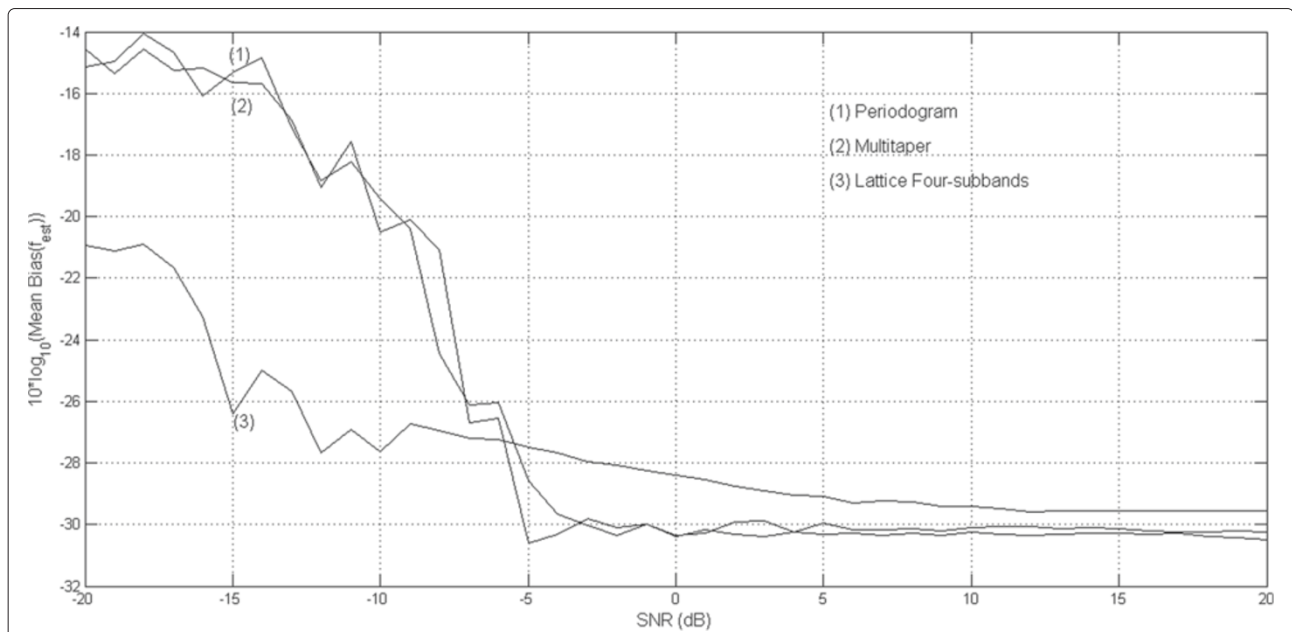


Figure 36 Comparison of the proposed method with the Periodogram and multitaper methods by means of mean bias versus SNR plots for estimating the frequencies of two cisoids embedded in colored noise.

multitaper in Figure 33, and the Capon, APES, IAA methods in Figure 34. In both of these figures, it can be observed that the proposed lattice in four-subbands performs much better than the nonparametric methods throughout the complete SNR range due to the whitening property of subband implementations.

In Figure 35, we present the mean bias curves, which demonstrate the relatively better performance of subband

implementations especially in the low-SNR region. In Figures 36 and 37, we compare the mean bias curves of the nonparametric methods with the proposed method using four-subbands. In both of these figures, there is a performance difference in favor of the four-subband lattice implementation in the SNR region ($-20 \text{ dB} \leq \text{SNR} \leq -5 \text{ dB}$), on the other hand, in the rest of the SNR region, the performance of the proposed lattice method

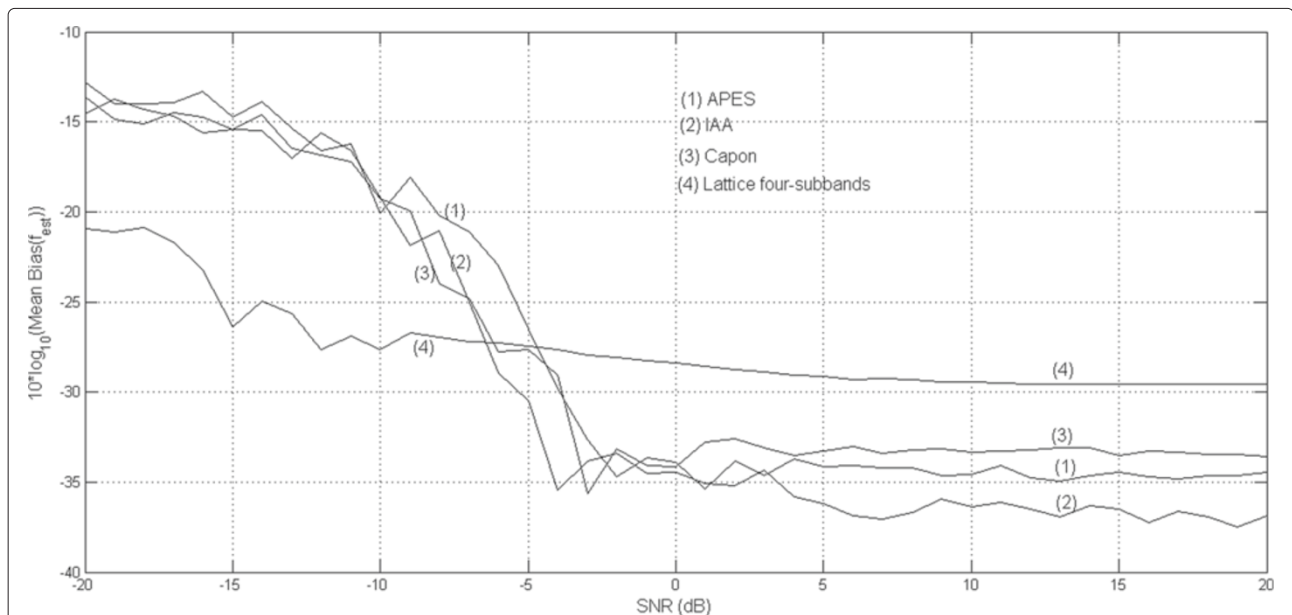
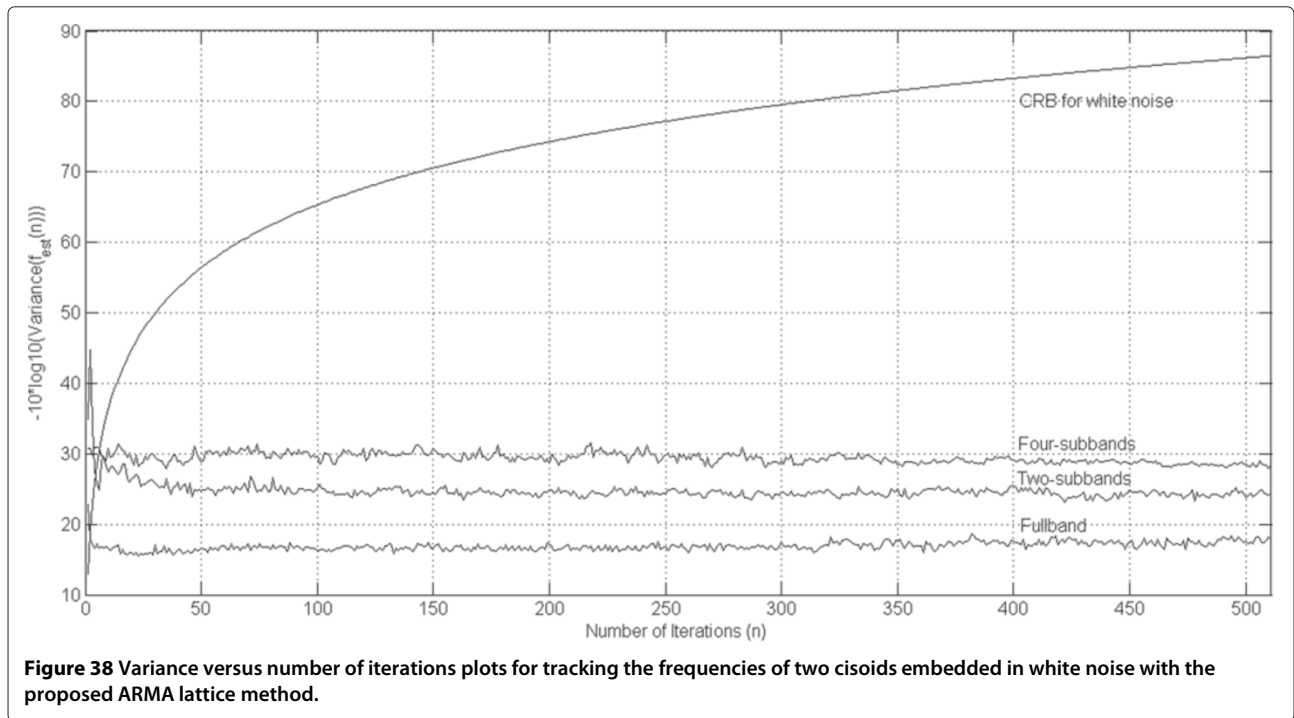


Figure 37 Comparison of the proposed method with the Capon, APES, and IAA methods by means of mean bias versus SNR plots for estimating the frequencies of two cisoids embedded in colored noise.

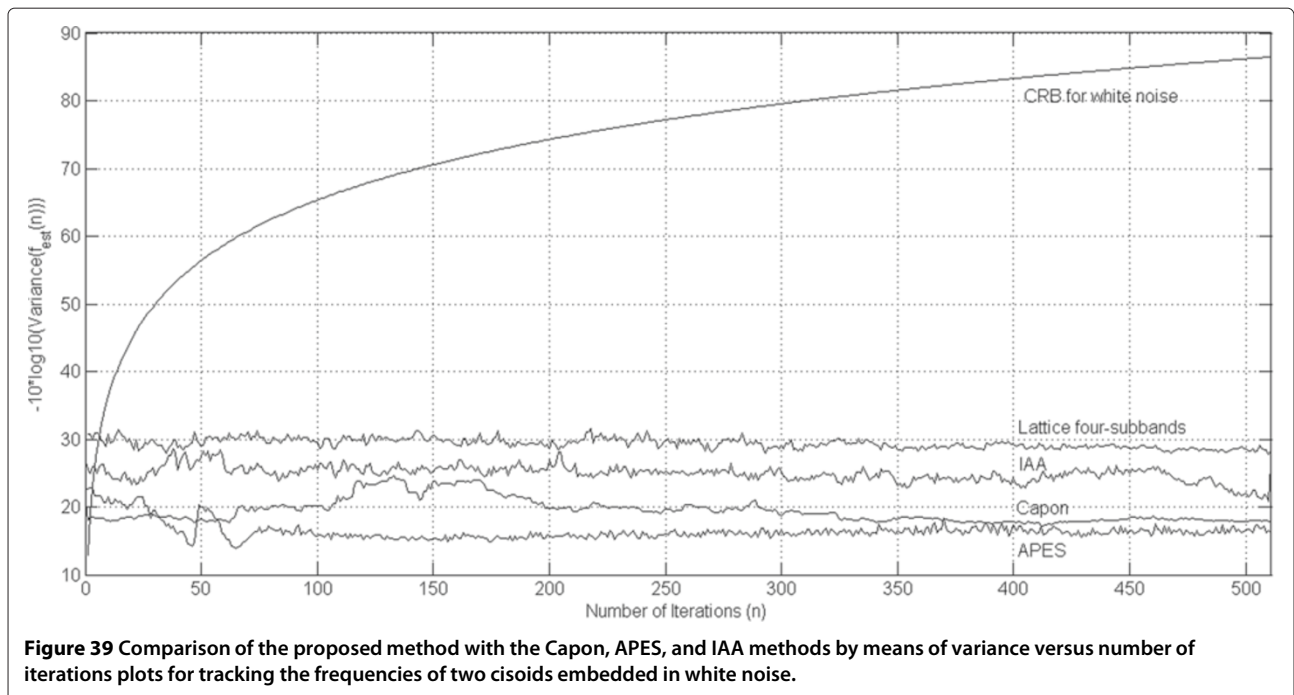


in four-subbands is worse than those of the Periodogram, multitaper, Capon, APES, and IAA methods.

6.2.2 Nonstationary signal case

We present the variance and mean bias of frequency estimation versus number of iterations plots when estimating the randomly time-varying frequencies of two cisoids

embedded in white noise ($SNR_1 = SNR_2 = -15$ dB) for the proposed lattice method in Figures 38 and 39 respectively. Therefore, the time series in this experiment were generated by using Equation (90) with frequencies replaced with the randomly time varying frequencies in Equation (95). The plots presented in Figures 38 and 39 are for one of the frequencies in order to improve the



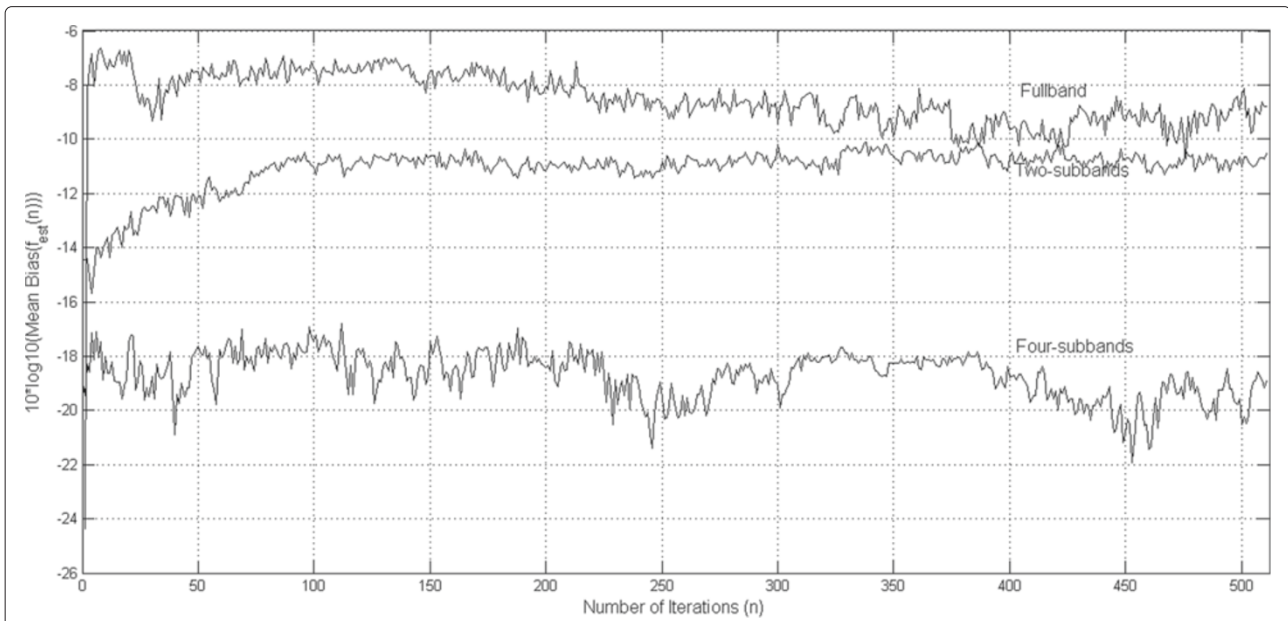


Figure 40 Mean bias versus number of iterations plots for tracking the frequencies of two cisoids embedded in white noise with the proposed ARMA lattice method.

readability, and they also display that the tracking performance improves as we increase the number of subbands. We think that the number of subbands in the proposed method plays a similar role to the pole contraction factor, ρ , in [8,9], and frequency tracking with the proposed lattice method in fullband, two-subband, and four-subband cases can be likened to increasing pole contraction factors in notch filtering. An approximate way of corresponding

the number of subbands to pole contraction factor can be by using the normalized bandwidth of complex notches, which is given as $BW = (1 - \rho)$ in [8,9]. Accordingly, a typical pole contraction factor of $\rho = 0.95$, and therefore the normalized notch bandwidth of $BW = 0.05$ corresponds to using ten-subbands in our method. In Figures 40 and 41, we compare the variance and mean bias versus number of iterations curves of the proposed lattice method in

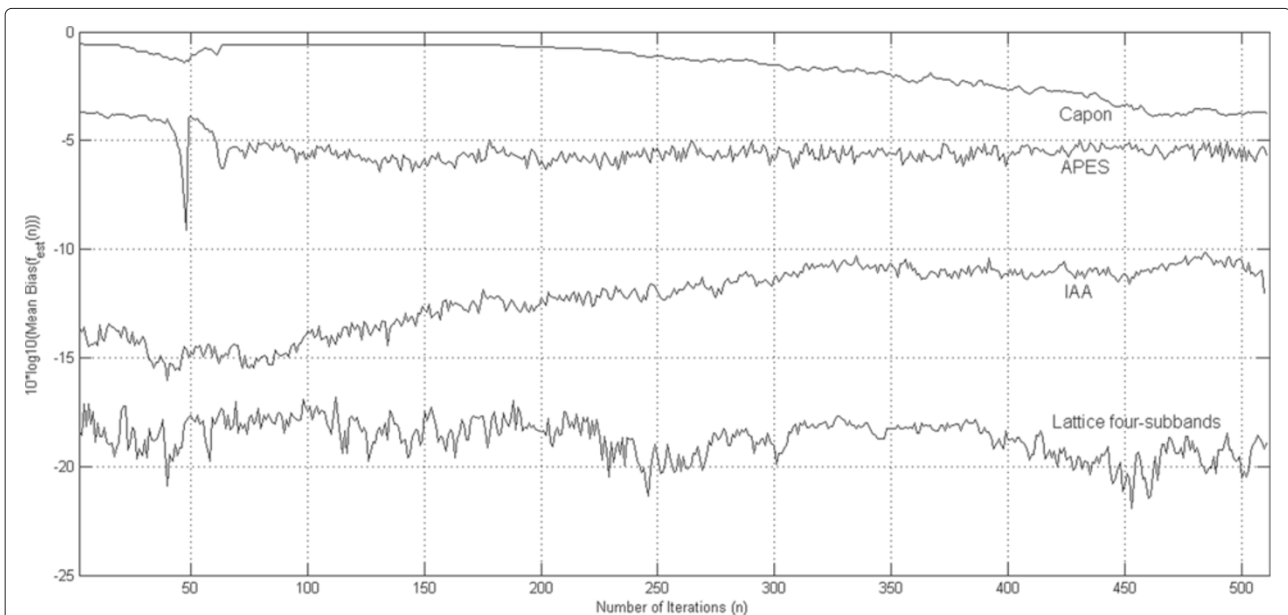


Figure 41 Comparison of the proposed method with the Capon, APES, and IAA methods by means of mean bias versus number of iterations plots for tracking the frequencies of two cisoids embedded in white noise.

four-subbands with the adaptive nonparametric methods (Capon, APES, and IAA), and demonstrate that the variance and mean bias performances of the proposed lattice four-subband method are better those of the adaptive Capon, APES, and IAA methods, and it is also considered that the better performance of the proposed method in such a low-SNR condition is mainly due to the SNR improvement effect of subband filtering.

7 Conclusions

A novel lattice method for adaptive ARMA spectrum estimation in subbands has been presented. The proposed method is different from previous methods such that it first transforms the ARMA spectrum estimation problem in subbands into an equivalent, multichannel prediction filtering problem, and then completely orthogonalizes the multichannel input signal using SPMLs. In order to convert lattice reflection coefficients to process parameters, the method includes a new Levinson–Durbin type multichannel conversion algorithm specially developed for SPMLs. The fullband spectrum estimation is a special form of the proposed subbands spectrum estimation method, and also gives rise to a novel two-channel prediction filter implementation.

The advantages of the method are that it can dynamically parameterize all subbands at once using a single filter structure, which is modular, regular, order-recursive, and therefore suitable for VLSI implementations such as developing both programable DSP chips and dedicated system on chip solutions; its inherently good numerical properties due to the avoidance of matrix inversions; and scalar only operations. The other strengths of the method are applicability to uniform as well as nonuniform subband filtering, and also to on-line and off-line implementations.

A detailed computational analysis has been provided by comparing the complexity of the proposed method with those of adaptive fast RLS transversal prediction filtering in subbands and the Periodogram method for different model orders and data lengths. Then, the complexity has also been compared against the multitaper, the Capon, APES, and IAA for one data length ($N = 128$), and it has been shown that the proposed method is computationally advantageous comparing to the multitaper and Capon methods for a range of lattice prediction filter orders, while it is computationally more advantageous than the APES method for a wide range of lattice prediction filter orders, and it is less complex than the IAA method for a very wide range of lattice prediction filter orders.

The simulation results demonstrate that the proposed subband spectrum estimation method has the SNR and frequency spacing improvement and whitening effect

advantages of a typical subband implementation. In stationary signal case, the variance and mean bias values versus SNR performance when estimating the frequencies of two cisoids in additive white and colored noise has been presented, and compared against the CRB for white noise as well as against the performances of the Periodogram, multitaper, Capon, APES, and IAA methods. In nonstationary signal cases, the performance has also been investigated, and compared against the Capon, APES, and IAA methods under deterministically and randomly time-varying signal conditions.

The performance results display that the proposed method improves the poor performance of parametric methods in low SNR and small frequency spacing, and colored noise conditions. It can also perform competitively with respect to the nonparametric methods in a computationally efficient manner. Consequently, it is considered that the proposed method with increasing number of subbands has good potential for cognitive radio spectrum sensing, radar and speech recognition tasks, and frequency estimation and tracking applications.

Competing interests

The author declares that he has no competing interests.

Acknowledgements

The author is grateful to the Editor Prof. Cassio G. Lopes and anonymous reviewers for their useful comments.

Received: 18 October 2012 Accepted: 4 January 2013

Published: 28 January 2013

References

1. S Rao, WA Pearlman, Analysis of linear prediction, coding, and spectral estimation from subbands. *IEEE Trans. Inf. Theory.* **142**(4), 1160–1178 (1996)
2. D Bonacci, C Mailhes, PM Djuric, in *Proceedings of the IEEE International Conference Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 6. Improving frequency resolution for correlation-based spectral estimation methods using subband decomposition, (May 2003), pp. 329–332
3. D Bonacci, C Mailhes, JY Tournet, in *Proceedings of the IEEE International Conference Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 4. Subband decomposition using multichannel AR spectral estimation, (May 2005), pp. 409–412
4. S Haykin, Radar signal processing. *IEEE ASSP Mag.* **2**(2), 2–18 (1985)
5. R Carriere, RL Moses, in *Proceedings of the IEEE National Radar Conference. Autoregressive moving average modeling of radar target signatures*, (20–21 April 1988), pp. 225–229
6. I Lim, BG Lee, Lossless pole-zero modeling of speech signals. *IEEE Trans. Speech Audio Process.* **1**(3), 269–276 (1993)
7. I Lim, BG Lee, Lossy pole-zero modeling of speech signals. *IEEE Trans. Speech Audio Process.* **4**(2), 81–88 (1996)
8. A Nehorai, A minimal parameter adaptive notch filter with constrained poles and zeros. *IEEE Trans. Acoust. Speech, Signal Process.* **33**(4), 983–996 (1985)
9. P Händel, A Nehorai, Tracking analysis of an adaptive notch filter with constrained poles and zeros. *IEEE Trans. Signal Process.* **42**(2), 281–291 (1994)
10. KB Eom, R Chellappa, Noncooperative target classification using hierarchical modeling of high-range resolution radar signatures. *IEEE Trans. Signal Process.* **45**(9), 2318–2327 (1997)
11. L Yong-Xiang, L Xiang, Z Zhao-Wen, in *Proceedings of the IEEE International Conference on Neural Networks and Signal Processing*, vol. 2. Modeling of

- multirate signal in radar target recognition, (14–17 Dec 2003), pp. 1604–1606
12. HK Kwan, M Wang, in *Proceedings of the IEEE International Conference on Neural Networks and Signal Processing*, vol. 2. ARMA lattice model for speech analysis and synthesis, (14–17 Dec 2003), pp. 912–915
 13. P Tichavsky, P Höndel, Two algorithms for adaptive retrieval of slowly time-varying multiple cioids in noise. *IEEE Signal Process.* **43**(5), 1116–1127 (1995)
 14. PA Regalia, An improved lattice-based adaptive IIR notch filter. *IEEE Signal Process.* **39**(9), 2124–2128 (1991)
 15. MR Petraglia, SK Mitra, J Szczupak, Adaptive sinusoid detection using IIR notch filters and multirate techniques. *IEEE Trans. Circuits Syst. II. Analog and Digital Signal Process.* **41**(11), 709–717 (1994)
 16. Y Prudat, J-M Vesin, Multi-signal extension of adaptive frequency tracking algorithms. *Signal Process.* **89**, 963–973 (2009)
 17. H-Yu Liu, RY Yen, Effective adaptive iteration algorithm for frequency tracking and channel estimation in OFDM systems. *IEEE Trans. Veh. Technol.* **59**(4), 2093–2097 (2010)
 18. F Sandberg, M Stridh, L Sörnmo, Frequency tracking of atrial fibrillation using hidden markov models. *IEEE Trans. Biomed. Eng.* **55**(2), 502–511 (2008)
 19. G Evangelopoulos, P Maragos, Multiband modulation energy tracking for noisy speech detection. *IEEE Trans. Audio Speech Lang. Process.* **14**(6), 2024–2038 (2006)
 20. J Piskorowski, Suppressing harmonic powerline interference using multiple-notch filtering methods with improved transient behavior. *Measurement* (2012). doi:10.1016/j.measurement.2012.03.004
 21. L Tan, J Jiang, Novel adaptive IIR filter for frequency estimation and tracking. *IEEE Signal Process. Mag.* **26**(6), 186–189 (2009)
 22. J Ma, GY Li, BH Juang, Signal processing in cognitive radio. *Proc. IEEE.* **97**(5), 805–823 (2009)
 23. E Axell, G Leus, EG Larsson, HV Poor, Spectrum sensing for cognitive radio. *IEEE Signal Process. Mag.* **29**(3), 101–116 (2012)
 24. SL Marple, *Digital Spectral Analysis with Applications*. (Prentice Hall, Englewood Cliffs, NJ, 1987)
 25. F Ling, JG Proakis, A generalized multichannel least squares lattice algorithm based on sequential processing stages. *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-32**(2), 381–389 (1984)
 26. H Lev-ari, Modular architectures for adaptive multichannel lattice algorithms. *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-35**(4), 543–552 (1987)
 27. G-O Glentis, N Kalouptsidis, A highly modular adaptive lattice algorithm for multichannel least squares filtering. *Signal Process.* **46**, 47–55 (1995)
 28. PS Lewis, QR-based algorithms for multichannel adaptive least squares lattice filters. *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-38**(3), 421–431 (1990)
 29. B Yang, A QR multichannel least squares lattice algorithm for adaptive nonlinear filtering. *Int. J. Electron. Commun. (AEÜ)*. **49**(4), 171–182 (1995)
 30. AA Rontogiannis, S Theodoridis. *IEEE Signal Process.* **46**(11), 2862–2876 (1998)
 31. J Gomes, VAN Barroso, Array-based QR-RLS multichannel lattice filtering. *IEEE Signal Process.* **56**(8), 3510–3522 (2008)
 32. GA Glentis, N Kalouptsidis, Fast adaptive algorithms for multichannel filtering and system identification. *IEEE Signal Process.* **40**(10), 2433–2457 (1992)
 33. DTM Slock, L Chisci, H Lev-Ari, T Kailath, Modular and numerically stable fast transversal filters for multichannel and multiexperiment RLS. *IEEE Signal Process.* **40**(4), 784–802 (1992)
 34. PA Regalia, MG Bellanger, On the duality between fast QR methods and lattice methods in least squares adaptive filtering. *IEEE Trans. Signal Process.* **39**(4), 879–891 (1991)
 35. F Ling, Givens rotation based least squares lattice and related algorithms. *IEEE Trans. Signal Process.* **39**(7), 1541–1551 (1991)
 36. F Ling, A recursive modified Gram-Schmidt algorithm for least-squares estimation. *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-34**(4), 829–835 (1986)
 37. MT Ozden, AH Kayran, Decision feedback equalisation with complete lattice orthogonalisation. *Electron. Lett.* **37**(14), 923–926 (2001)
 38. MT Ozden, AH Kayran, Adaptive multichannel decision feedback equalization for volterra type nonlinear communication channels. *Int. J. Electron. Commun. (AEÜ)*. **62**(6), 430–437 (2008)
 39. R Woods, J McAllister, G Lightbody, Y Yi, *FPGA-Based Implementation of Signal Processing Systems*. (Wiley, New York, 2008)
 40. J Li, P Stoica, An adaptive filtering approach to spectral estimation and SAR imaging. *IEEE Trans. Signal Process.* **44**(6), 1469–1483 (1996)
 41. P Stoica, R Moses, *Spectral Analysis of Signals*. (Prentice Hall, Englewood Cliffs, NJ, 2005)
 42. T Yardibi, J Li, P Stoica, M Xue, AB Baggeroer, Source localization and sensing: a nonparametric iterative adaptive approach based on weighted least squares. *IEEE Trans. Aerosp. Electron. Syst.* **46**(1), 425–442 (2010)
 43. AB Thomson, Spectrum estimation and harmonic analysis. *Proc. IEEE.* **70**, 1055–1096 (1982)
 44. S Haykin, DJ Thomson, JH Reed, Spectrum sensing for cognitive radio. *Proc. IEEE.* **97**(5), 849–877 (2009)
 45. B Farhang-Boroujeny, Filter bank spectrum sensing for cognitive radios. *IEEE Trans. Signal Process.* **56**(5), 1801–1811 (2008)
 46. GO Glentis, Efficient algorithms for adaptive Capon and APES spectral estimation. *IEEE Trans. Signal Process.* **58**(1), 84–96 (2010)
 47. GO Glentis, A Jakobsson, Efficient implementation of iterative adaptive approach spectral techniques. *IEEE Trans. Signal Process.* **59**(9), 4154–4167 (2011)
 48. S Haykin, *Adaptive Filter Theory*, 4th edn. (Prentice Hall, Englewood Cliffs, NJ, 2004)
 49. P Strobach, New forms of Levinson and Schur algorithms. *IEEE Signal Process. Mag.* **8**(1), 12–36 (1991)
 50. SH Ardalan, LJ Faber, A fast ARMA transversal RLS filter algorithm. *IEEE Trans. Acoust. Speech, Signal Process.* **ASSP-36**(3), 349–358 (1998)
 51. WT Cochran, JW Cooley, DL Favin, HD Helms, RA Kaelnel, WW Lang, GC Maling, DE Nelson, CM Rader, PD Welch, What is the fast Fourier transform? *Proc. IEEE.* **55**(10), 1664–1674 (1967)
 52. Y-P Lin, VDD Vaidyanathan, A Kaiser window approach for the design of prototype filters of cosine modulated filterbanks. *IEEE Signal Process. Lett.* **5**(6), 132–134 (1998)
 53. GO Glentis, A Jakobsson, Time-recursive IAA spectral estimation. *IEEE Trans. Signal Process. Lett.* **18**(2), 111–114 (2011)
 54. PMT Broersen, S de Waele, Time series analysis in a frequency subband. *IEEE Trans. Instrum. Meas.* **52**(4), 1054–1060 (2003)
 55. SM Kay, *Modern Spectral Estimation: Theory and Application*. (Prentice Hall, Englewood Cliffs, NJ, 1987)

doi:10.1186/1687-6180-2013-9

Cite this article as: Ozden: Adaptive multichannel sequential lattice prediction filtering method for ARMA spectrum estimation in subbands. *EURASIP Journal on Advances in Signal Processing* 2013 **2013**:9.

Submit your manuscript to a SpringerOpen® journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com