ATTAINABLE ERROR BOUNDS IN MULTIRATE ADAPTIVE LOSSLESS FIR FILTERS

Phillip A. Regalia and Dong-Yan Huang

Institut National des Télécommunications
Département Signal et Image
9, rue Charles Fourier
91011 Evry cedex France

ABSTRACT

We consider the problem of adaptively optimizing a two-channel lossless FIR filter bank, which finds application in subband coding or wavelet signal analysis. Instead of using a gradient descent procedure—with its inherent problem of possible convergence to local minima—we consider two eigenstructure algorithms. Both algorithms feature *a priori* bounds on the output error variance at any convergent point, and based on simulations lead to solutions that lie acceptably close to a global minimum point of an output error cost function.

1. INTRODUCTION

The role of lossless filter banks in wavelet or subband signal analysis is by now well recognized [6]. The basic two-channel maximally decimated filter bank is shown in Figure 1, comprising M rotation angles if the filter length is M.

A common problem in many applications is to synthesize a filter bank in the form of Figure 1, such that the variance of the output $y_2(n)$ is "small," possibly minimized with respect to some criterion. If the spectral density of the input signal prior to decimation is available, then the design problem is essentially deterministic, and solutions are easily approached [4].

In real-time applications, adaptive designs for the rotation angles $\theta_k(n)$ may form an attractive alternative to costly off-line optimization methods. The most obvious approach is to use a gradient descent procedure applied to the cost function $E[y_2^2(n)]$; adaptive algorithm design parallels [5] closely, leading to order M^2 computations per time sample. An inherent drawback of such an approach is that the cost function $E[y_2^2(n)]$ is nonquadratic in the rotation angles θ_k, such that local minima may result. Although the global minimum will certainly lead to the "best" solution, local minima can yield suboptimal performance.

Avoiding local minima requires either global search methods, or adaptation algorithms that do not follow the negative gradient of the cost function $E[y_2^2(n)]$. The latter possibility is explored in this paper, leading to two algorithms of order M complexity. Both algorithms seek to embed $E[y_2^2(n)]$ as the extremal eigenvalue of a covariance matrix; if successful, *a priori* bounds on $E[y_2^2(n)]$ after convergence may be developed, versus the filter length M. Simulations are included to illustrate asymptotic performance.

2. PROBLEM STRUCTURE

We begin with the two-channel lossless FIR filter of Figure 1, using M rotation angles $\Theta = [\theta_1, \ldots, \theta_M]$. This system may be described as

$$\begin{bmatrix} \mathbf{x}(n+1) \\ \mathbf{y}(n) \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{A}(\Theta) & \mathbf{B}(\Theta) \\ \mathbf{C}(\Theta) & \mathbf{D}(\Theta) \end{bmatrix}}_{\triangleq\, \mathbf{Q}(\Theta)} \begin{bmatrix} \mathbf{x}(n) \\ \mathbf{u}(n) \end{bmatrix}, \qquad (1)$$

where $\mathbf{x}(\cdot) = [x_1(\cdot), \ldots, x_{M-1}(\cdot)]^t$ is the state vector, $\mathbf{u}(n) = \left[\begin{smallmatrix} u_1(n) \\ u_2(n) \end{smallmatrix}\right]$ is the input vector, and $\mathbf{y}(n) = \left[\begin{smallmatrix} y_1(n) \\ y_2(n) \end{smallmatrix}\right]$ is the output vector. The input vector derives from a scalar process by way of subsampling, i.e.,

$$u_1(n) = u(2n), \qquad u_2(n) = u(2n-1).$$

We assume that $\{u(\cdot)\}$ is a zero mean, stationary second-order process.

The matrix $\mathbf{Q}(\Theta)$ in (1) is the cascade of rotations, and so is always orthogonal. This implies that the overall transfer matrix $\mathbf{V}(z)$, as in

$$\begin{bmatrix} y_1(n) \\ y_2(n) \end{bmatrix} = \mathbf{V}(z) \begin{bmatrix} u_1(n) \\ u_2(n) \end{bmatrix},$$

is para-unitary, i.e.,

$$\mathbf{V}(z)\, \mathbf{V}^t(z^{-1}) = \mathbf{V}^t(z^{-1})\, \mathbf{V}(z) = \mathbf{I}_2, \quad \text{for all } z,$$

irrespective of the rotation angles $\Theta = \{\theta_k\}$. A standard objective is to adjust the rotation angles according to the input signal so as to force $E[y_2^2(n)]$ to be "small".

Let us introduce the input spectral density matrix

$$\mathcal{S}_u(z) = \sum_{k=-\infty}^{\infty} E[\mathbf{u}(n)\,\mathbf{u}^t(n-k)]z^{-k}, \quad |z| = 1.$$

Since $\{\mathbf{u}(\cdot)\}$ derives from a scalar process $\{u(\cdot)\}$ by way of subsampling, one may show [3] that

$$\mathcal{S}_u(z) = \begin{bmatrix} S_e(z) & S_o(z) \\ z^{-1}S_o(z) & S_e(z) \end{bmatrix}$$

in which

$$\begin{aligned} S_e(z) &= \cdots + r_4 z^2 + r_2 z + r_0 + r_2 z^{-1} + r_4 z^{-2} + \cdots \\ S_o(z) &= \cdots + r_3 z^2 + r_1 z + r_1 + r_3 z^{-1} + r_5 z^{-2} + \cdots \end{aligned}$$

are the polyphase components from the input spectral density function prior to subsampling:

$$S(z) = \sum_{k=-\infty}^{\infty} r_k\, z^{-k} \quad \text{with } r_k = E[u(n)\,u(n-k)].$$

In the same way, the output power spectral density is

$$\begin{aligned} \mathcal{S}_y(z) &= \sum_{k=-\infty}^{\infty} E[\mathbf{y}(n)\,\mathbf{y}^t(n-k)]\,z^{-k} \\ &= \mathbf{V}(z) \begin{bmatrix} S_e(z) & S_o(z) \\ z^{-1}S_o(z) & S_e(z) \end{bmatrix} \mathbf{V}^t(z^{-1}). \end{aligned}$$

Remark: If $S_o(z)$ vanishes, i.e., $r_k = 0$ for k odd, then $\mathcal{S}_u(z) = S_e(z)\,\mathbf{I}_2$, so that

$$\mathcal{S}_y(z) = S_e(z)\,\mathbf{V}(z)\,\mathbf{V}^t(z^{-1}) = S_e(z)\,\mathbf{I}_2$$

as well. This case gives, in particular,

$$E[y_2^2(n)] = \frac{1}{2\pi}\int_{-\pi}^{\pi} S_e(e^{j\omega})\,d\omega = r_0, \qquad (2)$$

irrespective of which para-unitary $\mathbf{V}(z)$ we use. Accordingly, if $S_o(z)$ is negligible compared to r_0, the cost function $E[y_2^2(n)]$ versus Θ is fairly flat, and reduces to a constant function r_0 whenever $S_o(z)$ vanishes. This simple fact will be exploited in Section 4.

3. FIRST ALGORITHM

Instead of using a gradient descent on the cost function $E[y_2^2(n)]$, consider the algorithm

$$\begin{bmatrix} \theta_1(n+1) \\ \vdots \\ \theta_M(n+1) \end{bmatrix} = \begin{bmatrix} \theta_1(n) \\ \vdots \\ \theta_M(n) \end{bmatrix} - \mu\, y_2(n)\,\Gamma(n) \begin{bmatrix} \mathbf{x}(n+1) \\ y_1(n) \end{bmatrix} \qquad (3)$$

in which $\Gamma = \mathrm{diag}[\gamma_1, \ldots, \gamma_M]$ with

$$\gamma_M(n) = 1, \quad \gamma_k(n) = \gamma_{k+1}(n)\cos\theta_k(n).$$

The recursion (3) is closely related to one proposed earlier in rational subspace estimation [2].

For slow adaptation, the convergence properties of this algorithm are weakly linked with an associated differential equation of the form

$$\frac{d\Theta}{dt} = -\Gamma(\Theta)\,E\left\{ \begin{bmatrix} \mathbf{x}(n+1) \\ y_1(n) \end{bmatrix} y_2(n) \middle| \Theta \right\} \qquad (4)$$

in which the right-hand side expectation is evaluated for constant parameters Θ. The result varies with Θ, which thus defines a function of Θ which drives the differential $d\Theta/dt$. A given point in the Θ parameter space is, for slow adaptation, a convergent point in mean of (3) if and only if this same value of Θ is an attractive stationary point of the differential equation (4).

In order to understand the stationary points of this algorithm, introduce the covariance matrix

$$\mathbf{K}(\Theta) \triangleq E\left\{ \begin{bmatrix} \mathbf{x}(n) \\ \mathbf{u}(n) \end{bmatrix} [\mathbf{x}^t(n) \quad \mathbf{u}^t(n)] \middle| \Theta \right\}.$$

If we partition $\mathbf{Q}(\Theta)$ from (1) into the form

$$\mathbf{Q}(\Theta) = \begin{bmatrix} \mathbf{Q}_1(\Theta) \\ \mathbf{q}_2(\Theta) \end{bmatrix} \begin{matrix} \}\, M \text{ rows} \\ \}\, 1 \text{ row} \end{matrix}$$

then

$$E\left\{ \begin{bmatrix} \mathbf{x}(n+1) \\ y_1(n) \end{bmatrix} y_2(n) \middle| \Theta \right\} = \mathbf{Q}_1(\Theta)\,\mathbf{K}(\Theta)\,\mathbf{q}_2^t(\Theta).$$

A stationary point of the algorithm, say Θ_*, is attained if and only if the column vector $\mathbf{K}(\Theta_*)\,\mathbf{q}_2^t(\Theta_*)$ is orthogonal to the M rows of $\mathbf{Q}_1(\Theta_*)$, i.e.,

$$\mathbf{K}(\Theta_*)\,\mathbf{q}_2^t(\Theta_*) = \lambda\,\mathbf{q}_2^t(\Theta_*). \qquad (5)$$

The eigenvalue λ is simply

$$\lambda = \mathbf{q}_2(\Theta_*)\,\mathbf{K}(\Theta_*)\,\mathbf{q}_2^t(\Theta_*) = E[y_2^2(n)].$$

In fact, we have:

Property 1. *At any convergent point (in mean) of algorithm (3),*

$$E[y_2^2(n)] = \lambda_{min}[\mathbf{K}(\Theta_*)] \leq \lambda_{M+1}[\mathbf{R}]$$

where $\mathbf{R}$ *is the input autocorrelation matrix*

$$\mathbf{R} = \begin{bmatrix} r_0 & r_1 & r_2 & \cdots & r_{2M-1} \\ r_1 & r_0 & r_1 & \cdots & r_{2M-2} \\ r_2 & r_1 & r_0 & \cdots & r_{2M-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{2M-1} & r_{2M-2} & r_{2M-3} & \cdots & r_0 \end{bmatrix}.$$

For the first equality, one may show that, because $\mathbf{Q}(\Theta)$ may be permuted to an orthogonal Hessenberg matrix, the rows of $\mathbf{Q}_1(\Theta)$ generate the derivatives of $\mathbf{q}_2(\Theta)$:

$$\Gamma(\Theta)\,\mathbf{Q}_1(\Theta) = \begin{bmatrix} \partial\mathbf{q}_2(\Theta)/\partial\theta_1 \\ \vdots \\ \partial\mathbf{q}_2(\Theta)/\partial\theta_M \end{bmatrix}.$$

The differential equation (4) then appears related to a Rayleigh quotient iteration, and the identical argument used in [2] shows that, at any convergent point, the eigenvalue λ from (5) must correspond to the smallest eigenvalue of the covariance matrix $\mathbf{K}(\Theta_*)$.

Although $\mathbf{K}(\Theta)$ varies with the rotation angles Θ, a uniform bound may be shown for its eigenvalues. To verify, it suffices to note that

$$\mathbf{K}(\Theta) = \underbrace{\begin{bmatrix} \bigcirc & \mathbf{B} & \mathbf{AB} & \cdots & \mathbf{A}^{M-1}\mathbf{B} \\ \mathbf{I}_2 & \bigcirc & \bigcirc & \cdots & \bigcirc \end{bmatrix}}_{\triangleq\, \mathcal{C}(\Theta)} \mathbf{R}\,\mathcal{C}^t(\Theta).$$

Since $\mathbf{A}(\Theta)$ and $\mathbf{B}(\Theta)$ are extracted from an orthogonal matrix $\mathbf{Q}(\Theta)$, it is easy to check that $\mathcal{C}^t(\Theta)\,\mathcal{C}(\Theta) = \mathbf{I}_{M+1}$ for all Θ. Accordingly, the Poincaré separation theorem [1] shows that, for all Θ,

$$\lambda_k[\mathbf{K}(\Theta)] \le \lambda_k[\mathbf{R}], \qquad k = 1, 2, \ldots, M{+}1,$$

where we assume the eigenvalues are set in descending order. We then observe that $\lambda_{min}[\mathbf{K}(\Theta_*)]$ is just $\lambda_{M+1}[\mathbf{K}(\Theta_*)]$, to obtain the final bound of Property 1.

One may easily find inputs for which the algorithm (3) cannot converge. For if the polyphase component $S_o(z)$ of the input spectral density is negligible compared to r_0, then $E[y_2^2(n)] \approx r_0$ for all Θ. But $\lambda_{M+1}[\mathbf{R}]$ may, according to the "even" spectral density function chosen, be considerably smaller than r_0. Property 1 then cannot apply. We remark that, in this case, the error surface is quite flat, such that little coding gain is available anyway.

4. SECOND ALGORITHM

Since the even-indexed correlation terms r_{2k} make no contribution to $E[y_2^2(n)]$, we develop an alternate algorithm which uses only the odd-indexed terms r_{2k+1}.

Let $\widehat{\mathbf{R}}$ be the autocorrelation matrix obtained by modulating the scalar process $\{u(\cdot)\}$ by $(-1)^n$:

$$\widehat{\mathbf{R}} = E\left\{\begin{bmatrix} u(2n) \\ -u(2n-1) \\ u(2n-2) \\ -u(2n-3) \\ \vdots \end{bmatrix} [\cdot]^t\right\} = E\left\{\begin{bmatrix} u_1(n) \\ -u_2(n) \\ u_1(n-1) \\ -u_2(n-1) \\ \vdots \end{bmatrix} [\cdot]^t\right\}$$

We can then see that

$$\begin{aligned} \mathbf{R}_e &= \tfrac{1}{2}[\mathbf{R} + \widehat{\mathbf{R}}] \\ \mathbf{R}_o &= \tfrac{1}{2}[\mathbf{R} - \widehat{\mathbf{R}}] \end{aligned}$$

are the matrices built from the even- and odd-indexed parts, respectively, of the input correlation coefficients. If, for example, $M = 2$, then we would have

$$\mathbf{R}_e = \begin{bmatrix} r_0 & 0 & r_2 & 0 \\ 0 & r_0 & 0 & r_2 \\ r_2 & 0 & r_0 & 0 \\ 0 & r_2 & 0 & r_0 \end{bmatrix}, \quad \mathbf{R}_o = \begin{bmatrix} 0 & r_1 & 0 & r_3 \\ r_1 & 0 & r_1 & 0 \\ 0 & r_1 & 0 & r_1 \\ r_3 & 0 & r_1 & 0 \end{bmatrix}$$

Property 2. *If λ is an eigenvalue of $\mathbf{R}_o$, so is $-\lambda$.*

For if we set $\xi = \mathbf{R}_o\,\eta$, then negating each even-indexed component of η is equivalent to negating each odd-indexed component of ξ. Now choose η as an eigenvector of $\mathbf{R}_o$; the remaining steps follow easily. ◇

Consider now applying the modulated input to the same lossless filter bank, as depicted in Figure 2. With hatted accents absorbed throughout, we see that

$$\widehat{\mathbf{K}}(\Theta) = \mathcal{C}(\Theta)\,\widehat{\mathbf{R}}\,\mathcal{C}^t(\Theta).$$

This allows us to introduce

$$\mathbf{K}_o(\Theta) \triangleq \tfrac{1}{2}[\mathbf{K}(\Theta) - \widehat{\mathbf{K}}(\Theta)] = \mathcal{C}(\Theta)\,\mathbf{R}_o\,\mathcal{C}^t(\Theta),$$

in which only the odd-indexed terms of the input autocorrelation intervene. This suggest the following adaptation algorithm:

$$\begin{aligned} \Theta(n{+}1) &= \Theta(n) - \frac{\mu}{2}\Gamma(\Theta) \qquad (6) \\ &\times \left(y_2(n)\begin{bmatrix} \mathbf{x}(n{+}1) \\ y_1(n) \end{bmatrix} - \hat{y}_2(n)\begin{bmatrix} \hat{\mathbf{x}}(n{+}1) \\ \hat{y}_1(n) \end{bmatrix}\right). \end{aligned}$$

The same supporting arguments for Property 1 show that, at any convergent point Θ_* of this algorithm,

$$\mathbf{K}_o(\Theta_*)\,\mathbf{q}_2^t(\Theta_*) = \lambda_{M+1}[\mathbf{K}_o(\Theta_*)]\,\mathbf{q}_2^t(\Theta_*),$$

at which point

$$\begin{aligned} \tfrac{1}{2}\big(E[y_2^2(n)] - E[\hat{y}_2^2(n)]\big) &= \lambda_{M+1}[\mathbf{K}_o(\Theta_*)] \\ &\le \lambda_{M+1}[\mathbf{R}_o]. \end{aligned}$$

From Property 2, the eigenvalues $\lambda_1, \ldots, \lambda_M$ of $\mathbf{R}_o$ are positive, while the eigenvalues $\lambda_{M+1}, \ldots, \lambda_{2M}$ are negative. Accordingly, $\lambda_{M+1}[\mathbf{K}_o(\Theta)]$ is also negative.

We can note next that $\frac{1}{2}\big(E[y_2^2(n)] + E[\hat{y}_2^2(n)]\big)$ depends only on the even-indexed correlation terms of the input. It may thus be reduced to an integral of the form (2), giving the value r_0. Solving for $E[y_2^2(n)]$ then yields:

Property 3. *At any convergent point of algorithm (6),*

$$\begin{aligned} E[y_2^2(n)] &= r_0 + \lambda_{min}[\mathbf{K}_o(\Theta_*)] \\ &\leq r_0 + \lambda_{M+1}[\mathbf{R}_o] = r_0 - \lambda_M[\mathbf{R}_o]. \end{aligned}$$

Since both algorithms (3) and (6) are seeking extremal eigenequations, the convergence speed is limited by the eigenvalue separation of $\mathbf{K}(\Theta)$ or $\mathbf{K}_o(\Theta)$. Well separated eigenvalues lead to rapid convergence, while poorly separated eigenvalues lead to slow convergence. By a simple extension of Property 2, the matrix $\mathbf{K}_o(\Theta)$ will in general have better eigenvalue separation than $\mathbf{K}(\Theta)$, which in simulations leads to more rapid convergence of algorithm (6) compared to (3).

5. SIMULATION RESULTS

The input signal $\{u(\cdot)\}$ was obtained from the output of the filter

$$\text{poles}\begin{cases} -0.3 \\ 0.75 \pm j0.5809 \\ -0.5 \pm j0.8307 \end{cases} \quad \text{zeros}\begin{cases} 1.05 \\ -0.2 \\ 0.9 \pm j0.4472 \end{cases}$$

when driven by white noise, normalized to yield $r_0 = 1$.

Choosing $M = 4$, the global minimum of $E[y_2^2(n)]$ was found at

$$\begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} = \begin{bmatrix} 0.5704 \\ 0.4755 \\ -0.2115 \\ -0.1556 \end{bmatrix} \quad \text{giving} \quad E[y_2^2(n)] = 0.0219$$

The *a priori* bound for the first algorithm (3) is

$$E[y_2^2(n)] \leq \lambda_5[\mathbf{R}] = 0.0760$$

The algorithm exhibited a unique convergent point at

$$\begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} = \begin{bmatrix} 0.6521 \\ 0.4580 \\ -0.1875 \\ -0.1298 \end{bmatrix} \quad \text{giving} \quad E[y_2^2(n)] = 0.0228$$

For the second algorithm (6), the *a priori* bound is far more conservative:

$$E[y_2^2(n)] \leq r_0 - \lambda_4[\mathbf{R}_o] = 0.8998$$

The convergent point is nonetheless quite satisfactory:

$$\begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} = \begin{bmatrix} 0.6318 \\ 0.4529 \\ -0.2281 \\ -0.1185 \end{bmatrix} \quad \text{giving} \quad E[y_2^2(n)] = 0.0237$$

Both algorithms have, for this example, found an acceptable neighborhood of the global minimum.

6. CONCLUDING REMARKS

Two alternative algorithms were presented for adaptive filter bank optimization in multirate processing. The complexity of either algorithm is order M computations per iteration, yielding an improvement over the order M^2 complexity which would accompany a gradient descent approach.

Both algorithms, when convergent, yield *a priori* bounds on $E[y_2^2(n)]$ in terms of the input signal eigenstructure, although simulations indicate that these bounds are sometimes conservative.

7. REFERENCES

[1] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge Univ. Press, 1985.

[2] P. A. Regalia and Ph. Loubaton, "Rational subspace estimation using adaptive lossless filters," *IEEE Trans. Signal Processing*, vol. 40, pp. 2392–2405, Oct. 1992.

[3] V. P. Sathe and P. P. Vaidyanathan, "Effects of multirate systems on the statistical properties of random signals," *IEEE Trans. Signal Processing*, vol. 41, pp. 131–146, Jan. 1993.

[4] M. K. Tsatsanis and G. B. Giannakis, "Principal component filter banks for optimal wavelet analysis," report UVA/192444B/EE91/127, University of Virginia, 1991.

[5] M. Tummala and S. R. Parker, "A new efficient adaptive cascade lattice structure," *IEEE Trans. Circuits and Systems*, vol. 34, pp. 707–711, 1987.

[6] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*, Prentice-Hall, Englewood Cliffs, NJ, 1992.

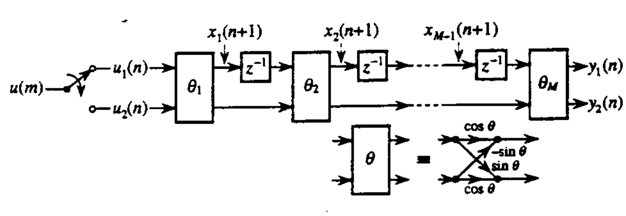

Figure 1: Two-channel lossless FIR filter bank.

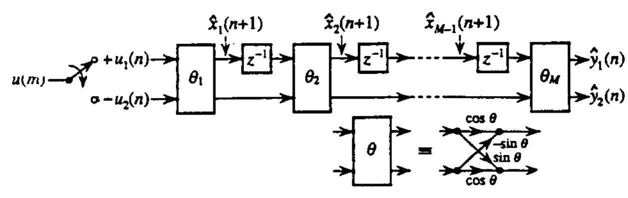

Figure 2: Modulated filter bank.