Def Sci J, Vol 35, No 3, July 1985, pp 287-294

Adaptive Lattice Algorithms for Passive Array Data

N. L. M. MURUKUTLA

Research and Training Unit for Navigational Electronics, Osmania University, Hyderabad.

T. S. DURRANI

Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, U.K.

Abstract. This paper reports the development of an algorithm for the processing of data from an array of broad-band sensors using lattice type processor. The problem is to enhance the look-direction signal in the presence of spatially distributed interference sources and sensor self noise by employing a multi-channel processor subject to the constraint that it has a desired response for look-direction signals. The multi-channel lattice algorithm proposed here possess stage by stage decoupling, and do not involve an arbitrary size of the step length, unlike conventional tapped-delay-line algorithms.

1. Introduction

The requirement for processing of data from multi-element sensors exists in numerous and diverse fields, such as in passive sonar signal processing of hydrophone array data; for sidelobe suppression in antenna arrays; in seismic processing for profiling multi-layered media.

Passive sonar array data which represents incident acoustic signals in broad-band noise, usually covers a wide frequency range. After pre-processing the data, adaptive processing techniques are usually employed for locating the source direction. For narrow-band (single frequency) data, a directional spectra accomplishes a mapping of the magnitude of the radiated power impinging on the array from various directions in space. For broad-band data, the problem is to enhance the weak look-direction signal in the presence of spatially distributed interference sources and sensor noise. Usually this problem is addressed by employing a multi-channel processor subject to the constraint that it has a desired frequency response for look-direction signals.

Techniques for dealing with the narrow-band data have been dealt extensively by Gabriel[1] where the filter coefficients are computed for generating the directional wave spectra. This problem is not addressed in this paper. The conventional techniques of dealing with the broad-band data are concerned with the use of tapped-delay-line filters (Fig. 1) with adaptive tap weights for nulling out interference source signals[2]. While these techniques have the advantage of simplicity and comparitive ease of computing, they tend to be slow in convergence rates, and are susceptible to the choice of arbitrary step size, and to the associated ill-conditioning of the covariance matrices of the array data.

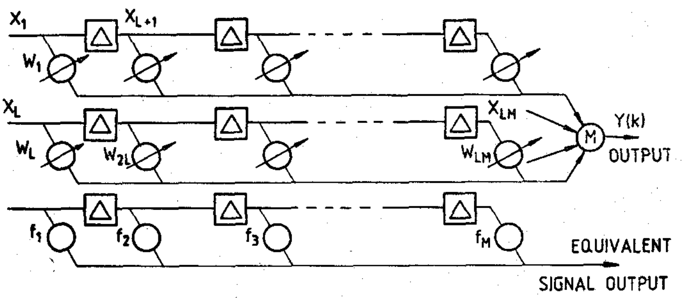

Figure 1. Broad-band array processor and equivalent processor for signal coming from look-direction.

Cantoni & Godara's[2] algorithm is a combination of Frost[3] & Ljung *et al.*[4] recursive least mean square estimation algorithms for adaptive beam-forming for a Frost type processor with constraints. The recursive algorithm of Ljung *et al.*[4] exploits the shift property of the array processor states to reduce the computational burden. Recursive least squares algorithms usually converge more rapidly than gradient based algorithms but at a price. The gradient based algorithms are not only slower in convergence on a per sample (or iteration) basis but also the samples used need to be independent, so that the real time interval between iterations is longer than for recursive least squares. Even though the algorithm of Cantoni & Godara converges faster than Frost's algorithm, the algorithm is highly susceptible to the step size.

Lattice structures, introduced by Itakura[5] for speech analysis, have several attractive features which make them potentially preferable to tapped-delay-line filters. Their modular structures, stability, properties, successive orthogonalization and decoupling of their residuals of each order, and superior convergence behaviour are well known for the lattice algorithms[6]. In this paper the use of adaptive lattices is extended to the processing of data arising from an array of sensors, and an algorithm is proposed, which can be conveniently implemented as constrained multi-channel lattices. The philosophy employed here is that: data from an array of sensors is fed to multi-channel lattices with adaptive reflection coefficients, the output power at each multi-channel lattice stage is minimized, subject to the constraint that for look-direction signals, which appear as identical signals on all input sensors, the multi-channel lattices collapse to a single-channel lattice with known reflection coefficients. The known reflection coefficients ensure that the processor offers a chosen frequency response to the look-direction signals.

2. Conventional Form of Constrained Processor

As a preamble to the constrained lattice processor, consider the conventional tapped delay line filter. Let the data sequence $\{X(k), k = 1, 2, \ldots N\}$ be the observations from an array of L sensors and M tap weights (Fig. 1) such that $X(k)$ is an $(LM \times 1)$ vector. Since the look-direction frequency response is fixed by the M a-priori constraints, the cost criterion for minimizing the total array output power is given by :

$$J = \tfrac{1}{2}\, \underline{W}^T R_{xx} \underline{W} + \lambda^T (C^T \underline{W} - \underline{f}) \tag{1}$$

where

R_{xx} is the covariance matrix of the input data

C is the constraint matrix

$\underline{\lambda}$ is the Lagrange multiplier

and $\underline{f}$ is the constraint matrix

$\underline{W}_{opt}$ is found by minimizing J with respect to $\underline{W}$ and satisfying the constraints, this leads to

$$\underline{W}_{opt}, = R_{xx}^{-1}\, C(C^T R_{xx} C)^{-1} \underline{f} \tag{2}$$

A hill climbing algorithm for $\underline{W}$ is given by

$$\underline{W}(k+1) = P\,[\underline{W}(k) - \mu R_{xx} \underline{W}(k)] + P' \underline{f} \tag{3a}$$

$$P = I - C(C^T C)^{-1}\, C^T \tag{3b}$$

$$P' = C(C^T C)^{-1} \tag{3c}$$

$$\mu = \alpha / \sigma_x^2\,(k) \tag{3d}$$

and

$$\sigma_x^2\,(k) = (1 - \alpha)\, \sigma_x^2\,(k-1) + \alpha \underline{X}^T(k)\, \underline{X}(k) \tag{3e}$$

and $0 < \alpha > 1$

3. Constrained Lattice Algorithm

Consider the data sequence $\{\underline{X}(k), k = 1, 2, \ldots N\}$ as the equally spaced observation from an array of L sensors, such that each $\underline{X}(k)$ is a $(L \times 1)$ vector. Fig. 2(a) represents the direct form of the lattice processor for array data, where each stage is identical. $F_m(k)$ and $\underline{G}_m(k)$, are the outputs of each stage, which may be referred to as the forward and backward residuals. K_m and L_m may be called the forward and backward reflection coefficients and each is an $(L \times L)$ matrix.

To generate a constrained lattice processor, we recast the linear weight constraints, such as $C^T \underline{W} - \underline{f}$ of Eqn. (1) into constraints on the reflection coefficient matrix.

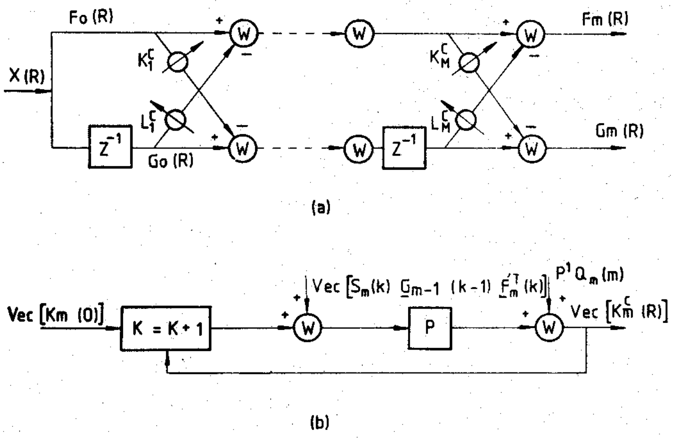

Figure 2. Constrained lattice algorithm processor.

The constraints to be applied to the multi-channel lattice processor are :

$$\frac{\underline{h}^T K_m \underline{h}}{\underline{h}^T \underline{h}} = \frac{\underline{h}^T L_m \underline{h}}{\underline{h}^T \underline{h}} = a_m(m) \tag{4}$$

$$m = 1, 2, \ldots, M$$

where $\underline{h}^T = [1, 1, \ldots, 1, 1]$ is a $(1 \times L)$ unit vector and $a_m(m)$ is the constraint on the mth stage, and it is the equivalent scalar (known) reflection coefficients of this filter which ensure desired response of the processor to the look-direction signals.

The approach is to minimise the output power at each stage $\{E[\underline{F}_m^T(k)\ \underline{F}_m(k)]$ and $E[\underline{G}_m^T(k)\ \underline{G}_m(k)]\}$ subject to the constraint that $\underline{h}^T K_m \underline{h}$ and $\underline{h}^T L_m \underline{h}$ reduce to predetermined scalar constants $a_m(m)$. The stage by stage output minimization involves the following criterion (for the mth stage)

$$J_m = \underset{K_m, L_m}{\text{Min}}\ \frac{1}{2} \sum_{t=1}^{k} \{\underline{F}_m^T(t)\, \underline{F}_m(t) + \underline{G}_m^T(t)\, \underline{G}_m(t)\}$$

$$+ \lambda_m(k) \left(\frac{\underline{h}^T K_m(k) \underline{h}}{\underline{h}^T \underline{h}} - a_m(m) \right) + \mu_m(k) \left(\frac{\underline{h}^T L_m(k)\, \underline{h}}{\underline{h}^T \underline{h}} - a_m(m) \right) \tag{5}$$

Here the constraints representing reflection coefficients of the desired single channel filter $a_m(m)$ are embedded into the cost criterion. The two sets of scalar Lagrange multipliers $\{\lambda_m, \mu_m, m = 1, 2, \ldots, M\}$. Note that such a constraint needs to be applied across each order (stage) of the lattice processor. However, as the optimization in the lattice processor progresses from stage to stage, it is necessary to consider any mth stage as given here. The mth stage prediction error residuals are given by Jones[7] as

$$\underline{F}_m(k) = \underline{F}_{m-1}(k) - K_m G_{m-1}(k-1) \tag{6}$$

and

$$\underline{G}_m(k) = G_{m-1}(k-1) - L_m \underline{F}_{m-1}(k) \tag{7}$$

The cost criterion J_m in Eqn. (5) is minimized from the time sample $t = 1$ to $t = k$. This implies that the forward error residual $\underline{F}_m(t)$ is updated on the basis of kth value of the reflection coefficient matrix K_m. Replacing K_m in Eqn. (6) by $K_m(k)$, and substituting from $F_m(t)$, leads to

$$J_m = \underset{K_m}{\text{Min}}\ \tfrac{1}{2} \sum_{t=1}^{k} \{\underline{F}_{m-1}(t) - K_m(k)\, \underline{G}_{m-1}(t-1)\}^T$$
$$\times \{\underline{F}_{m-1}(t) - K_m(k)\, \underline{G}_{m-1}(t-1)\} + \lambda_m(k) \left[\frac{\underline{h}^T K_m(k)\, \underline{h}}{\underline{h}^T \underline{h}} - a_m(m)\right] \quad (8)$$

Applying Kronecker products to Eqn. (8) and differentiating it with respect to Vec $(K_m(k))$ and arranging it results in

$$\{\sum_{t=1}^{k} (\underline{G}_{m-1}(t-1)\, \underline{G}^T_{m-1}\, (t-1) \otimes I)\}\ \text{Vec}\,(K_m(k)$$
$$= \{\sum_{t=1}^{k} (G_{m-1}(t-1)\, \underline{F}^T_{m-1}(t) \otimes I)\ \text{Vec}\, I\}$$
$$- \lambda_m(k)\, (\underline{h}\underline{h}^T \otimes I)\ \text{Vec}\,(I)/\underline{h}^T\underline{h} \quad (9)$$

where $\otimes$ denotes Kronecker products

Defining

$$(S_m(k) \otimes I)^{-1} = \sum_{t=1}^{k} (\underline{G}_{m-1}(t-1)\, \underline{G}^T_{m-1}\, (t-1) \otimes I) \quad (10)$$

Applying Eqn. (10) in Eqn. (9) leads to

$$(S_m(k) \otimes I)^{-1}\ \text{Vec}\,(K_m(k)) = \{\sum_{t=1}^{k} (\underline{G}_{m-1}(t-1)\, \underline{F}^T_{m-1}\, (t) \otimes I)\ \text{Vec}\,(I)\}$$
$$+ (\underline{G}_{m-1}(k-1)\, \underline{F}^T_{m-1}(k) \otimes I)\ \text{Vec}\, I$$
$$- \lambda_m(k)\, (\underline{h}\underline{h}^T \otimes I)\ \text{Vec}\,(I)/\underline{h}^T\underline{h} \quad (11)$$

modifying Eqn. (11) leads to (here the same term is added and subtracted),

$$(S_m(k) \otimes I)^{-1}\ \text{Vec}\,(K_m(k)) = (S_m(k) \otimes I)^{-1}\ \text{Vec}\,(K_m(k-1))$$
$$- (\underline{G}_{m-1}(K-1)\, \underline{F}'^T_m\, (k) \otimes I).\ \text{Vec}\, I$$
$$- (\lambda_m(k) - \lambda_m(k-1))\, \underline{h}\underline{h}^T \otimes I)\ \text{Vec}\,(I)/\underline{h}^T\underline{h} \quad (12)$$

where $$(\underline{F}'_m(k) \otimes I)\ \text{Vec}\,(I) = (\underline{F}'_{m-1}\,(k) \otimes I)\ \text{Vec}\,(I)$$
$$- (\underline{G}^T_{m-1}\,(k-1) \otimes I)\ \text{Vec}\,(K_{m-1}\,(k-1) \quad (13)$$

premultiplying Eqn. (12) by $(S_m(k) \otimes I)$ and applying the constraint $(\underline{h}^T \otimes \underline{h}^T)$ Vec $(K_m(k))/\underline{h}^T\underline{h} = a_m(m)$, and solving it we have

$$\text{Vec}\,(K_m(k)) = P\,[\text{Vec}\,(K_m(k-1)) + (S_m(k) \otimes I)\,(\underline{G}_{m-1}(k-1) \times (\underline{F}_m^{'T}(k) \otimes I)\,\text{Vec}\,I] + P'\,a_m(m) \tag{14}$$

where

$$P = I - (S_m(k) \otimes I)\,(\underline{h}\underline{h}^T \otimes I)\,\text{Vec}\,I\,\{(\underline{h}^T \otimes \underline{h}^T)\,(S_m(k) \otimes I) \times (\underline{h}\underline{h}^T \otimes I)\,\text{Vec}\,I\}^{-1}\,(\underline{h}^T \otimes \underline{h}^T) \tag{15}$$

and

$$P' = (S_m(k) \otimes I)\,(\underline{h}\underline{h}^T \otimes I)\,\text{Vec}\,I\,\{(\underline{h}^T \otimes \underline{h}^T)\,(S_m(k) \otimes I) \times (\underline{h}\underline{h}^T \otimes I)\,\text{Vec}\,I\}^{-1}\,(\underline{h}^T\underline{h}) \tag{16}$$

From above it may be seen that

$$\frac{(\underline{h}^T \otimes \underline{h}^T}{\underline{h}^T\underline{h}}\,.\,P = O, \text{ and } (\underline{h}^T \otimes \underline{h}^T)\,P' = I$$

Hence

$$\frac{\underline{h}^T K_m \underline{h}}{\underline{h}^T\underline{h}} = \frac{(\underline{h}^T \otimes \underline{h}^T)}{\underline{h}^T\underline{h}}\,\text{Vec}\,(K_m(k)) = \frac{(\underline{h}^T \otimes \underline{h}^T)}{\underline{h}^T\underline{h}}\,P'\,a_m(m) = a_m(m)$$

Satisfy the desired constraint.

Also from Eqn. (11)

$$\begin{aligned}(S_m(k) \otimes I)^{-1} &= \{\sum_{t=1}^{k=1} (\underline{G}_{m-1}(t-1)\,\underline{G}_{m-1}^T(t-1) \otimes I)\} \\ &\quad + (\underline{G}_{m-1}(k-1)\,\underline{G}_{m-1}^T(k-1) \otimes I) \\ &= (S_m(k-1) \otimes I)^{-1} + (\underline{G}_{m-1}(k-1)\,\underline{G}_{m-1}^T(k-1 \otimes I)\end{aligned} \tag{17}$$

Applying the matrix inversion lemma to Eqn. (17)

$$\begin{aligned}(S_m(k) \otimes I) = (S_m(k-1) \otimes I) - (S_m(k-1) \otimes I)\,(\underline{G}_{m-1}(k-1) \otimes I) \\ \times [I + (\underline{G}_{m-1}^T(k-1) \otimes I)\,(S_m(k-1) \otimes I)\,(\underline{G}_{m-1}(k-1) \\ \times \otimes I)]^{-1}\,(\underline{G}_{m-1}^T(k-1) \otimes I)\,(S_m(k-1) \otimes I)\end{aligned} \tag{18}$$

Similarly, the backward reflection coefficient may be obtained by minimizing Eqn. (5) with respect to Vec $(L_m(k))$, we have

$$\text{Vec}\,(L_m(k)) = Q\,[\text{Vec}\,(L_m(k-1)) + (T_m(k) \otimes I)\,\{\underline{F}_{m-1}(k) \times \underline{G}_m^{'T}(k) \otimes I\}\,\text{Vec}\,(I)] + Q'\,a_m(m) \tag{19}$$

where

$$(\underline{G}'_m(k) \otimes I) \text{ Vec } I = (\underline{G}_{m-1}(k-1) \otimes I) \text{ Vec } I - (\underline{F}_{m-1}(k) \otimes I) \text{ Vec } (L_m(k-1)) \quad (20)$$

$$Q = [I - (T_m(k) \otimes I)(\underline{h}\underline{h}^T \otimes I) \text{ Vec } I \{(\underline{h}^T \otimes \underline{h}^T)(T_m(k) \otimes I) \times (\underline{h}\underline{h}^T \otimes I) \text{ Vec } I\}^{-1} (\underline{h}^T \otimes \underline{h}^T) \quad (21)$$

$$Q' = (T_m(k) \otimes I)(\underline{h}\underline{h}^T \otimes I) \text{ Vec } I \{(\underline{h}^T \otimes \underline{h}^T)(T_m(k) \otimes I) \times (\underline{h}\underline{h}^T \otimes I) \text{ Vec } I\}^{-1} \underline{h}^T\underline{h} \quad (22)$$

and

$$(T_m(k) \otimes I) = (T_m(k-1) \otimes I) - (T_m(k-1) \otimes I)(\underline{F}_{m-1}(k) \otimes I \times [I + (\underline{F}^T_{m-1}(k) \otimes I)(T_m(k-1) \otimes I)(\underline{F}_{m-1}(k) \otimes I)]^{-1} \times (\underline{F}^T_{m-1}(k) \otimes I)(T_m(k-1) \otimes I) \quad (23)$$

The matrices P and Q represent $(L^2 \times L^2)$ projection operators, P' and Q' are projection vectors and $(S_m(k) \times I)^{-1}$ and $(T_m(k) \times I)^{-1}$ are instantaneous covariance estimates of forward and backward residual vectors $F_m(k)$ and $G_m(k)$. The projection operators are idempotent, in that matrices $P.P = P$ and $Q.Q = Q$. Eqns. (14) & (19) satisfy the constraints set up in Eqn. (4). Equation (14), (19), (6) and (7) represent an exact solution to the constrained optimization problem as posed in Eqn. (5), and yield a multi-channel lattice structure which minimizes the output power for each stage while maintaining a desired response for the look-direction signals. The computations are recursive both in time and in stage order. The exact solution requires the time updating of F_m, G_m, K_m, L_m, T_m and S_m at each stage. A processor which implements the above algorithm is given in Fig. 2.

4. Results

Both the conventional and lattice algorithms were employed to compute the constrained weights of the tapped-delay line-filter or the constrained lattice reflection coefficient matrices interatively. The weights or the reflection coefficients are used to compute the output power in the case of conventional algorithm and power at each stage of lattice for lattice algorithm. Both the curves are plotted on the same axis against number of iterations for the following example.

Example : A linear array with five equally spaced omnidirectional elements and four tapped delay-line sections is used. The look direction signal is arriving from the broad-side of the linear array. The interference sources are impinging on the array at —60° and 45° relative to the broad-side signal.

The desired look-direction signal response and the signal specifications are illustrated in Fig. 3. Fig. 4 shows the convergence characteristics of the residual noise

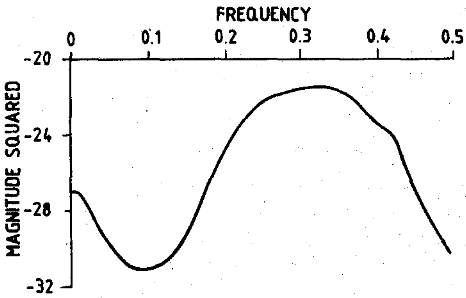

Figure 3. Frequency response of the processor in the look-direction.

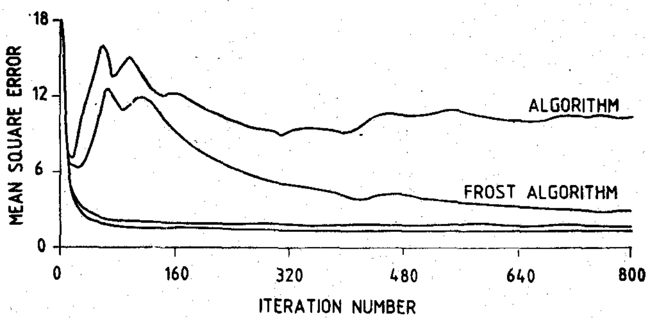

Figure 4. Comparison of convergence rates constrained frost algorithm and direct constrained adaptive lattice algorithm.

and interference source power at the output of various stages of lattice and the output of the array processor for the conventional algorithm. Note that the lattice algorithm converges much faster than Frost[3] algorithm. The convergence reaches the stable values at 50th iteration of the third stage of the lattice where as the conventional algorithm is still converging at $k = 800$. The first two stages of the lattice do not provide useful information regarding the convergence for the lattice algorithm.

References

1. Gabriel, W. F., *Proc. IEEE,* **68** (1980), 654–666.
2. Cantoni, A. & Godara, L. C., 'Fast Algorithms for Time Domain Adaptive Array Processing', Proc. of NATO Adv. Study Inst. on Underwater Acoust., Signal Processing, Copenhagen, Denmark, Aug. 1980.
3. Frost, O. L., *Proc. IEEE,* **60** (1972), 926–935.
4. Ljung, L., *et al., Int. J. of Control,* **27** (1978), 1–19.
5. Itakura, F. & Saito, S., 'Digital Filtering Techniques for Speech Analysis and Synthesis', Int. Congr. Acoustics, Budapest, Paper 25–C–1, (1971).
6. Makhol, J., *IEEE Trans. Acoust., Speech and Signal Processing,* **ASSP-26** (1978), 303–314.
7. Jones, R. H., 'Multivariate Maximum Entropy Spectral Analysis', Applied Time Series Analysis Symposium, Tulsa, OK, May (1976).