THE RECURSIVE DESIGN OF A SEISMIC ARRAY PROCESS

By

H. Kobayashi and P. D. Welch

IBM Research Center
Yorktown Heights, New York

Summary

Two methods of a recursive (iterative) synthesis of an array process are discussed: the method of steepest descent and the method of conjugate gradient with projection. These methods require no intermediate statistics such as the covariance matrix function or the cross-power spectral matrix, and therefore require less storage space than the conventional synthesis methods. Simulation results indicate that the convergence is so fast that a few iterations are enough from the practical viewpoint. Therefore these methods can save significant computation time as well.

I. Introduction

The model usually adopted in the problem of the detection and estimation of seismic events with an array of seismometers, is to assume that a signal due to a seismic event is common to all seismometers except for time delays $\tau_k = \vec{u} \cdot \vec{r}_k$ where $\vec{u}$ is the inverse phase velocity of the event and $\vec{r}_k$, the location of the k-th seismometer. In this model if we assume that the time delays are compensated so that the signal components are lined up, the input of the k-th seismometer is given by

$$x_k(t) = s(t) + n_k(t) \quad , \quad k=1, 2, \ldots, K \qquad (1.1)$$

where $s(t)$ is the unknown seismic signal and $n_k(t)$ represents all other disturbances in the k-th channel. Our interest lies in the case where the noise, $n_k(t)$, is highly correlated among seismometers. This will be the case, for example, when the main noise source is another interfering event, or when $n_k(t)$ is the first arrival of an event and $s(t)$ is a later arrival of the same event.

Let $\underline{w}(u)$ be a K-dimensional linear filter with finite duration $[-L_1, L_2]$ whose k-th component $w_k(u)$ is the impulse response of the k-th channel. Then the output, $y(t)$, is sum of these filter outputs:

$$y(t) = \sum_u \underline{w}^T(u)\, \underline{x}(t-u) \qquad (1.2)$$

Our purpose is to design the optimum $\underline{w}(u)$, based on the data taken over some fitting interval T_f, that minimizes the output noise power without distorting the signal $s(t)$: i.e., the criterion for optimality is to minimize

$$P_{out} = \frac{1}{N_f} \sum_{t \epsilon T_f} (y(t) - s(t))^2 \qquad (1.3)$$

under the constraint

$$\underline{w}^T(u) \cdot \underline{1} = \delta_{u,o} \quad , \quad u \epsilon [-L_1, L_2] \qquad (1.4)$$

which we call the "fidelity" constraint. N_f is the number of data points in the fitting interval, $\underline{1}$ is a K-dimensional vector whose entries are all unity. The fidelity constraint (1.4) is set in order to pass the signal component with no distortion. (Ref. 1)

Let us assume for the moment that the fitting interval T_f is chosen in such a way that the unknown signal $s(t)$ does not exist during T_f. Then the output noise power P_{out} of (1.3) is simply given by the following quadratic form:

$$P_{out} = \sum_u \sum_v \underline{w}^T(u)\, \underline{\underline{R}}(u,v)\, \underline{w}(u) \qquad (1.5)$$

where

$$\underline{\underline{R}}(u,v) = \frac{1}{N_f} \sum_{t \epsilon T_f} \underline{x}(t-u)\, \underline{x}^T(t-v) \qquad (1.6)$$

Then the optimum solution for $\underline{w}(u)$ has been obtained by Kelly and Levin [1] and Capon et. al. [2]:

The work reported was supported in part under contract AF 19-67-C-0198, sponsored by the Advanced Research Projects Agency, Department of Defense

$$\underline{w}_{opt}(u) = \sum_v \underline{\underline{R}}^{-1}(u,v)\, Q(v,0)\, \underline{1} \tag{1.7}$$

where

$$Q^{-1}(u,v) = \underline{1}^T\, \underline{\underline{R}}^{-1}(u,v)\, \underline{1} \tag{1.8}$$

If the unknown signal exists in the fitting interval Equation (1.3) becomes

$$P_{out} \cong \frac{1}{N_f} \sum_{t\epsilon T_f} y^2(t) - \frac{1}{N} \sum_{t\epsilon T_f} s^2(t) \tag{1.9}$$

where we have used the approximation

$$\sum_u \frac{1}{N_f} \sum_{t\epsilon T_f} \underline{w}^T(u)\, \underline{n}(t-u) \cdot s(t) \cong 0 \tag{1.10}$$

Since the second term of (1.9) does not depend on the filter, the minimization of P_{out} is equivalent to minimization of Eq. (1.5).

II. Recursive Design of the Array Process

In this section we will discuss two methods of a recursive (iterative) synthesis of an array process: the method of the steepest descent and the conjugate gradient method with projection. An iterative procedure generates a sequence of the filters $\{\underline{w}_i(u)\}$ which converges to the optimum filter $\underline{w}_{opt}(u)$ as i increases. These techniques lead to very efficient software implementations on general purpose computers. The major advantage of these methods is that the synthesis does not require the calculation of intermediate statistics such as the covariance matrix function or the cross-power spectral matrix. As a result storage requirements are kept to a minimum. Further, the optimum solution is achieved, starting from an arbitrary initial estimate by the repetitive use of the same formula. Hence, the processive program is not complicated and as simulation results indicate, that the convergence is so fast that a few iterations are enough from the practical viewpoint. Therefore, these methods possess significant computation time advantages.

2.1 Method of the Steepest Descent

The steepest descent method has been widely used in optimization problems and its application to estimation problems is discussed by Balakrishnan [3]. It is to be noted that there exists some similarity between the computation algorithms of the steepest descent method and of the stochastic approximation method, although the latter is applied to adaptive estimation or filtering when the input data is observed over a long interval and is assumed to be stationary or quasi-stationary. The application of the stochastic approximation method to the seismic array processor is discussed by Lacoss [4], to whom is due some of the mathematical formulation in the present section.

The minimization of P_{out} of (1.5) with the constraint (1.4) can be formulated as minimization of

$$J = \tfrac{1}{2}\sum\sum \underline{w}^T(u)\, \underline{\underline{R}}(u,v)\, \underline{w}(v) + \sum \lambda(u)\, \{\underline{w}^T(u)\, \underline{1} - \delta_{u,o}\} \tag{2.1}$$

where $\{\lambda(u)\}$'s are the Lagrangian coefficients. Then the gradient method provides the following recursive formula

$$\underline{w}_{i+1}(u) = \underline{w}_i(u) - \alpha_i \left[\frac{\partial J}{\partial \underline{w}(u)}\right]_{\underline{w} = \underline{w}_i} \tag{2.2}$$

where α_i is a positive scalar. On inserting (2.1) into (2.2) and using the constraint (1.4) we arrive at the following formula. :

$$\underline{w}_{i+1}(u) = \underline{w}_i(u) + \alpha_i\, \underline{p}_i(u) \tag{2.3}$$

where $\underline{p}_i(u)$ is the direction vector given by

$$\underline{p}_i(u) = -\underline{\underline{P}} \cdot \sum_v \underline{\underline{R}}(u,v)\, \underline{w}_i(v) \tag{2.4}$$

and $\underline{\underline{P}}$ is a K x K singular matrix of the form

$$\underline{\underline{P}} = (\underline{\underline{I}} - \frac{1}{K}\, \underline{1}\, \underline{1}^T), \tag{2.5}$$

The vectors $\underline{w}_i(u)$ and $\underline{p}_i(u)$ may be considered as points in an N-dimensional Euclidean space E^N, where $N = K(L_1 + L_2 + 1) = K \cdot L$. However, it is more convenient for the following discussion to regard E^N as a product of L copies of K-dimensional subspace E^K: $E^N = E^K \times E^K \times \ldots \times E^K$. Then the constraints (1.4) specify a (K-1) dimensional hyperplane in each E^K:

$$S_0 \;:\; \underline{w}^T(u) \cdot \underline{1} = 0 \quad \text{for } u \neq 0 \tag{2.6}$$

$$S_1 \;:\; (\underline{W}(u) - \frac{1}{K}\, \underline{1})^T\, \underline{1} = 0, \quad \text{for } u = 0 \tag{2.7}$$

The hyperplane S_0 contains the origin of E^K; the hyperplane S_1 contains a vector $\frac{1}{K}\underline{1}$ and is parallel to S_0. Then it is clear that $\underline{\underline{P}}$ of (2.5) is the projection operator from E^K into S_0. Similarly for any point $\underline{x}$ in E^K, its projection into S_1 is given by $\underline{\underline{P}}\,\underline{x} + \frac{1}{K} \cdot \underline{1}$.

As can be seen from the definition (2.4), the direction vector $\underline{p}_i(u)$ is in an N'-dimensional subspace $S_0 x \ldots x S_0 \triangleq \Sigma_0$, where $N' = (K-1) L$. If the initial choice $\underline{w}_0(u)$ is such that $\underline{w}_0(u) \in \Sigma_1 \triangleq S_0 x \ldots x S_0 \, x S_1 x \ldots x S_0$, then $\{\underline{w}_i(u)\}$ also lies in the dimensional subspace Σ_1 for all i. We choose the gain α_i in such a way that the next approximation is the point which minimizes J of (2.1) over all points on the line of action of the vector $\underline{p}_i(u)$ passing through $\underline{w}_i(u)$. This leads us to the following recursive formula:

Initialization: $\quad w_0 \in \Sigma_1$ (2.8a)

$$p_o = - P R w_0 \tag{2.8b}$$

For $i \geq 0$

$$\alpha_i = \frac{|p_i|^2}{(p_i, R p_i)} \tag{2.9a}$$

$$w_{i+1} = w_i + \alpha_i p_i \tag{2.9b}$$

$$p_{i+1} = p_i - \alpha_i P R p_i \tag{2.9c}$$

In Eqs. (2.8) and (2.9) we adopted the simplified notation R, p_i, w_i instead of $\underline{\underline{R}}(u,v)$ $\underline{p}_i(u)$, $\underline{w}_i(u)$ etc.. Equation (2.8b) should read as

$$\underline{p}_0(u) = - \underline{\underline{P}} \sum_v \underline{\underline{R}}(u,v) \, \underline{w}_0(v) \tag{2 10}$$

Similarly

$$|p_i|^2 = (p_i, p_i) = \sum_u \underline{p}_i^T(u) \, \underline{p}_i(u) \tag{2.11}$$

and

$$(p_i, Rp_i) = \sum_u \sum_v \underline{p}_i^T(u) \, \underline{\underline{R}}(u,v) \, \underline{p}_i(v) \tag{2.12}$$

From the formulae (2.9a) - (2.9c) it follows that $\underline{p}_{i+1}(u)$ is orthogonal to the previous direction vector $\underline{p}_i(u)$, i.e.:

$$(p_{i+1}, p_i) = \sum_u \underline{p}_{i+1}^T(u) \, \underline{p}_i(u) = 0 \tag{2.13}$$

The sequence of the output noise powers is monotone decreasing:

$$J_{i+1} - J_i = - \frac{|p_i|^4}{(p_i, Rp_i)} \leq 0 \tag{2.14}$$

Therefore J_0 can be written as

$$J_0 = \sum_{i=0}^{\infty} \frac{|p_i|^4}{(p_i, Rp_i)} \tag{2.15}$$

and $\{p_i\} \to 0$ as $i \to \infty$. Hence from Eq. (2.4), $\{\underline{w}_i(u)\}$ converges to $\underline{w}_{opt}(u)$ of (1.7).

Although the recursive formulae (2.8a) - (2.9c) appear to require one to compute the cross-correlation function $\underline{\underline{R}}(u,v)$, it can be written in the following way by substituting the definition (1.6).

Initialization: $\quad \underline{w}_0(u) \in \Sigma_1$ (2.16a)

$$\underline{p}_0(u) = - \langle \underline{x}(t-u) - x_{av}(t-u)\, \underline{1}, \; y_0(t) \rangle \tag{2.16b}$$

where

$$y_0(t) = \sum \underline{w}_0^T(u) \, \underline{x}(t-u) \tag{2.16c}$$

For $i \geq 0$

$$x_{av}(t) = \frac{1}{K} \underline{1}^T \cdot \underline{x}(t) \tag{2.16d}$$

$$q_i(t) = \sum \underline{p}_i^T(u) \, \underline{x}(t-u) \tag{2.17a}$$

$$\alpha_i = |p_i|^2 / \|q_i(t)\|^2 \tag{2.17b}$$

$$w_{i+1} = w_i + \alpha_i p_i \tag{2.17c}$$

$$\underline{p}_{i+1}(u) = \underline{p}_i(u) - \alpha_i \langle \underline{x}(t-u) - x_{av}(t-u)\, \underline{1}, \; q_i(t) \rangle \tag{2.17d}$$

where $\langle \; , \; \rangle$ is defined as

$$\langle f(t), g(t) \rangle = \frac{1}{N_f} \sum_{t \in T_f} f(t) \, g(t) \tag{2.18}$$

One may replace Eqs. (2.17b) and (2.17c) by the following:

$$\alpha_i = - \langle q_i(t), y_i(t) \rangle / \|q_i(t)\|^2 \tag{2.17c'}$$

and

$$y_{i+1}(t) = y_i(t) + \alpha_i q_i(t) \tag{2.17d'}$$

2.2 The Method of Conjugate Gradient with Projection

In the method of the steepest descent the gradient $\left[\frac{\partial I}{\partial w}\right]_i$ was used as the direction vector p_i to obtain the next approximation w_{i+1}. Although, this choice of p_i maximizes the instantaneous rate of change of J, it does not necessarily lead to the "best" approximation. Moreover the procedure does not yield the solution in a finite number of steps even though the dimensionality of the unknown $\underline{w}(u)$ is finite.

In the present section we will modify the fundamental conjugate gradient method [5-7] so as to be able to apply the method to our specific problem. The method of conjugate gradients was devised by Hestenes and Stiefel [5] to solve a system of simulataneous linear algebraic equations,

$$\underline{\underline{R}} \, \underline{w} = \underline{b} \tag{2.19}$$

where $\underline{\underline{R}}$ is an N x N positive definite matrix and

$\underline{w}$, an N x 1 vector of unknowns and $\underline{b}$ is an N x 1 vector of constants. This method is an N-step iterative one; i.e. the algorithm is applied to give successive approximations to the solution of the given linear systems and, if computations are done with complete accuracy, a solution is obtained after M iterations where $M \le N$ (the order of the system). Clearly the same algorithm can be applied to find $\underline{w}$ which minimizes the following function

$$f(\underline{w}) = \frac{1}{2} \underline{w}^T \underline{\underline{R}}\, \underline{w} - \underline{w}^T \underline{b} \tag{2.20}$$

By modifying the fundamental formula [5.6] we can obtain the following conjugate gradient iterative procedure leading to the minimization of the quadratic form (1.5) under the fidelity constraint.

Initialization: $w_0 \in \Sigma_1$ (2.21a)

$$p_0 = r_0 = - P R w_0 \tag{2.21b}$$

For $i \ge 0$

$$\alpha_i = |r_i|^2 / (p_i, Rp_i) \tag{2.22a}$$

$$w_{i+1} = w_i + \alpha_i p_i \tag{2.22b}$$

$$r_{i+1} = r_i - \alpha_i PRp_i \tag{2.22c}$$

$$\beta_i = |r_{i+1}|^2 / |r_i|^2 \tag{2.22d}$$

$$p_{i+1} = r_{i+1} + \beta_i p_i \tag{2.22e}$$

In place of (2.22a) and (2.22d) one may use

$$\alpha_i = (p_i, r_i)/(p_i, Rp_i) \qquad (2.22a)'$$

$$\beta_i = -(r_{i+1}, Rp_i)/(p_i, Rp_i) \qquad (2.22d)'$$

where P is the projection operator defined by the matrix (2.5).

Many relations hold among the quantities appearing in (2.21a) - (2.22e). The most important ones are

$$p_i \in \Sigma_0 \; , \; r_i \in \Sigma_0 \; , \; w_i \in \Sigma_1 \quad \text{for all } i \tag{2.23}$$

$$(r_i, r_j) = 0 \; , \; i \ne j \tag{2.24}$$

$$(p_i, Rp_j) = 0 \; , \; i \ne j \tag{2.25}$$

The projected gradients $\{r_i \,;\, i=0, \ldots, N'-1\}$ where $N' = (K-1)L$ form a set of orthogonal vectors in the subspace Σ_0 and $\{p_i \,;\, i=0, \ldots, N'-1\}$ form a set of "R-conjugate "or" R-orthogonal" vectors. Since R is positive definite, it follows that $(p_i, Rp_i) > 0$ and therefore $\{p_i \,;\, i=0, \ldots, N'-1\}$ are necessarily linearly independent and span the N'-dimensional subspace Σ_0. Since the solution vector w_{opt} and its initial guess w_0 are both in Σ_1, their difference is always in Σ_0 and is representable uniquely as a linear combination of the basis $\{p_i\}$. In fact, using the coefficients α_i of Eq. (2.22a) we have the relation

$$w_{opt} - w_0 = \sum_{i=0}^{N'-1} \alpha_i p_i \in \Sigma_0 \tag{2.26}$$

As in the case of the steepest descent method, the output noise power is decreased at each step of the iteration

$$J_{i+1} - J_i = -(p_i, r_i)^2 / (p_i, Rp_i)$$

$$= - |r_i|^4 / (p_i, Rp_i) \le 0 \tag{2.27}$$

Furthermore it can be shown that the following property in the fundamental gradient method can be carried over to our conjugate gradient method with projection: the approximation $w_j(u)$ is closer to the solution $w_{opt}(u)$ than $w_i(u)$, $i < j$, i.e. $|\underline{w}_{opt}(u) - \underline{w}_j(u)| \le |\underline{w}_{opt}(u) - \underline{w}_i(u)|$. The result indicates that if we stop the iterative process at any step, the last obtained approximation to the solution is the best, in the sense of being the closest to the true solution.

The iterative formula can again be written without resorting to the correlation function $\underline{\underline{R}}(u, v)$:

Initialization: $w_0 \in \Sigma_1$ (2.28a)

$$p_0(u) = r_0(u) = - \langle \underline{x}(t-u) - x_{av}(t-u)\, \underline{1},\; y_0(t) \rangle \tag{2.28b}$$

For $i \ge 0$

$$\alpha_i = |r_i|^2 / \|q_i(t)\|^2 \tag{2.29a}$$

$$w_{i+1} = w_i + \alpha_i p_i \tag{2.29b}$$

$$r_{i+1}(u) = \underline{r}_i(u) - \alpha_i \langle \underline{x}(t-u) - x_{av}(t-u)\, \underline{1},\; q_i(t) \rangle \tag{2.29c}$$

$$\beta_i = |r_{i+1}|^2 / |r_i|^2 \tag{2.29d}$$

$$p_{i+1} = r_{i+1} + \beta_i p_i \tag{2.29e}$$

where the functions $y_0(t)$, $x_{av}(t)$ and $q_i(t)$ are defined by (2.16c), (2.16d), and (2.17a).

Equations (2.29a) and (2.29b) can be replaced by

$$\alpha_i = - < q_i(t), \ y_i(t) > / \ \| q_i(t) \|^2 \qquad (2.29a)'$$

and

$$y_{i+1}(t) = y_i(t) + \alpha_i \ q_i(t) \qquad (2.29b)'$$

One may readily notice that if the coefficients $\{\beta_i\}$ in the iteration formula is set to zero, the conjugate gradient method reduces to the steepest descent method of the previous section.

III. Processing Requirements

The rationale of the iterative synthesis procedures is the efficient utilization of computer memory and processing time. In the present section we give some estimates of the memory requirements and the running time for the two methods described above.

The Steepest Descent Method: The input data $\underline{x}(t)$ must be stored in any method, and therefore, we exclude the space for $\underline{x}(t)$ in the following argument. The space for the processed output y(t) is also common to all methods and hence will be excluded here. The quantities w_i and p_i take KL words and $x_{av}(t)$ and $q_i(t)$ require N_f words, respectively. The memory requirement is thus $2(N_f + KL)$ words. However, if we do not require the processed output until the last iteration is over, $q_i(t)$ can be stored in the space alloted to y(t). In this respect the minimum memory requirement is

$$M_{sd} \cong (N_f + 2KL) \text{ words} \qquad (3.1)$$

As to the computation time, the majority is spent for the convolution sum to obtain $\underline{q}_i(t)$ and $\underline{p}_i(u)$ each of which takes about $KLN_f (\mu + \nu)$, where μ and ν are the MULTIPLY and ADD times in the computer in question. Therefore

$$T_{sd} = 2 \ KLN_f (\mu + \nu) \text{ seconds/iteration} \quad (3.2)$$

is the running time.

The Conjugate Gradient Method: The KL words for $\underline{r}_i(u)$ should be added to the quantities used in the method of the steepest descent. The total memory requirement is thus

$$M_{cg} \cong (N_f + 3KL) \text{ words} \qquad (3.3)$$

The increase of the running time over the steepest descent is $3KL(\mu + \nu)$ seconds per iteration due to the additional quantities β_i and $\underline{r}_i(u)$: Therefore

$$T_{cg} \cong (2KLN_f + 3KL) (\mu + \nu) \qquad (3.4)$$

is the running time.

V. Simulation Results

The iterative procedures outlined above were applied to the data from the Montana Large Seismic Array. Twenty-one center seismometers' data were used, i.e. K=21. The Longshot explosion's data (Oct. 29, '65) were superposed on the Kamchatka earthquake data (Apr. 8, '66) with a 30 second delay. Figure 1 shows processed outputs of the conjugate gradient method, where $L_1 = L_2 = 5$. The fitting interval is the first 25 seconds after the arrival of the Kamchatka earthquake, and $N_f = 500$ since the data sampling rate is 20 c/s. The interation starts from the simple beam-forming, i.e. $\underline{w}(u) = \frac{1}{K} \delta_{u,0} \underline{1}$, u= -5, ..., 0, ..., 5.

We notice that the interfering noise is suppressed drastically (more than 10 dB) by the first iteration, and an approximate solution attained after several iterations virtually satisfies the practical purpose.

Figure 2 is a plot of the output noise power vs. number of iterations for the steepest descent method as well as for the conjugate gradient method when filter lengths are L = 1 ($L_1 = L_2 = 0$), L = 11 ($L_1 = L_2 = 5$) and L = 21 ($L_1 = L_2 = 10$). In all cases the conjugate gradient (C-G) method shows a faster convergence than the steepest descent (S-D) method. We also synthesized filters in the frequency domain [2], where filter lengths of L = 8 and 16 were chosen so that the Fast Foruier Transform subroutine could be utilized. Reduction of the noise power output attained by those filters is also indicated in Figure 2, and is clearly less than the reduction achieved after several steps in iterative methods.

All programs were written in Fortran IV and an IBM 360 Model 67 digital computer was used for the simulation. The computer running time per iteration was:

	L = 1	L = 11	L = 21
S-D method	4.28 sec.	38.16 sec.	72.30 sec.
C-G method	5.07 sec.	46.00 sec.	87.27 sec.

On the other hand, the frequency domain synthesis takes 622.0 sec. for L = 8 and 1135 sec. for L = 16. Although the computation effort should be compared using programs written in a machine language to be precise, the simulation allows us to conclude that the iterative design provide an efficient way of synthesizing array processors.

Acknowledgment

The authors are indebted to G. H. Purdy for his programming support.

References

1. E. J. Kelly and M. J. Levin, "Signal Parameter Estimation for Seismometer Arrays", Mass. Inst. Tech., Lincoln Lab. Tech. Rept. 339, January 8, April, 1964.

2. J. Capon et. al., "Multidimensional Maximum-Likelihood Processing of a Large Aperture Seismic Array, " Proc. of IEEE, February, 1967, pp. 192-211.

3. A. V. Balakrishnan, "Filtering and Prediction Theory", in Communication Theory, ed. A. V. Balakrishnan, McGraw-Hill, 1968.

4. R. T. Lacoss, "Adaptive Combining of Wideband Array Data for Optimal Reception", IEEE Trans. on Geoscience Electronics, May, 1968, pp. 78-86.

5. M. Hestenes and E. Stiefel, "Methods of Conjugate Gradients for Solving Linear Systems", J. of Research of the Nat. Bur. Standards, December, 1952, pp. 409-436.

6. F. S. Beckman, "The Solution of Linear Equations by the Conjugate Gradient Method", in Mathematical Methods for Digital Computers, eds. A. Ralston and H. S. Wilf, Wiley, 1960.

7. R. Fletcher and M. J. D. Powell, "A Rapidly Convergent Descent Method for Minimization", The Computer Journ., Vol. 6, pp. 163-168, 1963.

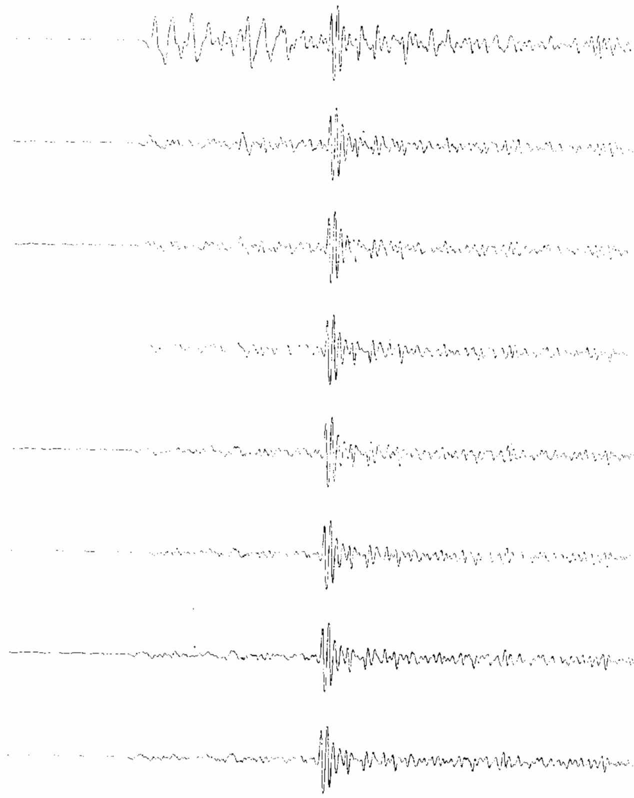

Figure 1. Processed Outputs of an Array Processor Designed by the Conjugate Gradient Method

($L_1 = L_2 = 5$, $N_f = 500$)

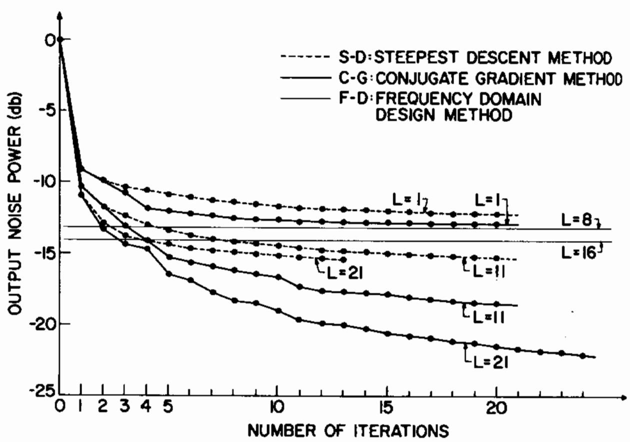

Figure 2. Output Noise Power vs. Number of Iterations