Iterative Synthesis Methods for a Seismic Array Processor

HISASHI KOBAYASHI, MEMBER, IEEE

Abstract—Two methods for the iterative synthesis of an array processor are discussed: the method of steepest descent and the method of conjugate gradients 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. A bound for the rate of convergence is obtained for these iterative procedures and it is shown that the convergence is geometric. The algorithms are then applied to seismic data of the Montana large aperture seismic array. Simulation results indicate that the convergence is so fast that a few iterations are enough from the practical viewpoint. Therefore, these methods can also 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 $\vec{r}_k$ is the location of the kth seismometer. In this model, if we assume that the time delays τ_k are compensated for, so that the signal components are lined up, the input of the kth seismometer is given by

$$x_k(t) = s(t) + n_k(t), \qquad k = 1, 2, \cdots, K \tag{1}$$

where $s(t)$ is the unknown seismic signal and $n_k(t)$ represents all other disturbances in the kth channel. Our interest lies in the case where the noise $n_k(t)$, is highly correlated among seismometers. This will be true, 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. Throughout the present paper the incoming data $x_k(t)$ are assumed to be time-discrete random processes, and hence our discussion is limited to the case in which a processor is a digital filter.

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

$$\begin{aligned} y(t) &= \sum_{u=-L_1}^{L_2} \sum_{k=1}^{K} w_k(u) x_k(t-u) \\ &= \sum_{u=-L_1}^{L_2} \boldsymbol{w}^T(u) \boldsymbol{x}(t-u) \end{aligned} \tag{2}$$

where $\boldsymbol{w}^T(u)$ is the transpose of the K-dimensional vector function $\boldsymbol{w}(u)$, and $\boldsymbol{x}(t)$ is a K-dimensional vector whose kth component is given by (1).

Our purpose is to design the optimum $\boldsymbol{w}(u)$, 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

$$P_{\text{out}} = \frac{1}{N_f} \sum_{t \in T_f} (y(t) - s(t))^2 \tag{3}$$

under the constraint

$$\boldsymbol{w}^T(u)\mathbf{1} = \delta_{u,0} = \begin{cases} 0, & u \neq 0, \quad -L_1 \leqq u \leqq L_2 \\ 1, & u = 0 \end{cases} \tag{4}$$

which we call the "fidelity" constraint. N_f is the number of data points in the fitting interval; $\mathbf{1}$ is a K-dimensional vector whose entries are all unity. The fidelity constraint (4) is set in order to pass the signal component with no distortion [1].

Let us assume, for the moment, that a 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 (3) is simply given by the following quadratic form:

$$P_{\text{out}} = \sum_{u=-L_1}^{L_2} \sum_{v=-L_1}^{L_2} \boldsymbol{w}^T(u) \bar{\boldsymbol{R}}(u, v) \boldsymbol{w}(v) \tag{5}$$

where

$$\bar{\boldsymbol{R}}(u, v) = \frac{1}{N_f} \sum_{t \in T_f} \boldsymbol{x}(t-u) \boldsymbol{x}^T(t-v), \qquad -L_1 \leqq u, \quad v \leqq L_2. \tag{6}$$

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

$$\boldsymbol{w}_{\text{opt}}(u) = \sum_{v=-L_1}^{L_2} \bar{\boldsymbol{R}}^{-1}(u, v) Q(v, 0)\mathbf{1}. \tag{7}$$

Manuscript received August 16, 1969; revised March 10, 1970. This paper was presented in part at the Third Annual Princeton Conference on Information Science and Systems, March 27–28, 1969 [10]. This work was supported in part by the Advanced Research Project Agency, Department of Defense, under Contract AF 19-67-C-0198.

The author is with the IBM Thomas J. Watson Research Center, Yorktown Heights, N. Y.

Here $\bar{R}^{-1}(u, v)$ is the inverse of the matrix correlation function $\bar{R}(u, v)$ of (6):

$$\sum_{v=-L_1}^{L_2} \bar{R}^{-1}(u, v)\bar{R}(v, t) = \bar{I}\cdot\delta_{u,t}, \quad -L_1 \leqq u,\ t \leqq L_2 \quad (8)$$

where $\bar{I}$ is $K\times K$ identity matrix. Function $Q(v, t)$ of (7) is defined by

$$Q^{-1}(u, v) = 1^T\bar{R}^{-1}(u, v)1 \quad (9a)$$

and

$$\sum_{v=-L_1}^{L_2} Q^{-1}(u, v)Q(v, t) = \delta_{u,t}, \quad -L_1 \leqq u,\ t \leqq L_2. \quad (9b)$$

If the unknown signal exists in the fitting interval, (3) becomes

$$P_{\text{out}} \cong \frac{1}{N_f}\sum_{t\in T_f} y^2(t) - \frac{1}{N_f}\sum_{t\in T_f} s^2(t) \quad (10)$$

where we have used the approximation

$$\frac{1}{N_f}\sum_{t\in T_f} \boldsymbol{w}^T(u)\boldsymbol{n}(t-u)\cdot s(t) \cong 0. \quad (11)$$

Since the second term of (10) does not depend on the filter, the minimization of P_{out} is equivalent to minimization of (5).

II. Iterative Design of an Array Processor

In this section we will discuss two methods for the iterative synthesis of an array processor: the method of steepest descent and the conjugate gradient method with projection. An iterative procedure generates a sequence of the filters $[\boldsymbol{w}_i(u)]$ which converges to the optimum filter $\boldsymbol{w}_{\text{opt}}(u)$ 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 processing program is not complicated and as simulation results indicate, the convergence is so fast that a few iterations are enough from the practical viewpoint. Therefore, these methods possess a significant computation time advantage.

A. 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 a close similarity between the computation algorithms of the steepest descent method as applied to the present paper 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. An application of the stochastic approximation method to adaptive antenna systems is reported by Widrow *et al.* [4], and the application to the seismic array processor is discussed by Lacoss [5], to whom is due some of the mathematical formulation in the present section.

The minimization of P_{out} of (5) with the constraint (4) can be formulated as minimization of

$$J = \sum_u\sum_v \boldsymbol{w}^T(u)\bar{R}(u, v)\boldsymbol{w}(v) + \sum_u \lambda(u)\{\boldsymbol{w}^T(u)1 - \delta_{u,0}\} \quad (12)$$

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

$$\boldsymbol{w}_{i+1}(u) = \boldsymbol{w}_i(u) - \frac{\alpha_i}{2}\left[\frac{\partial J}{\partial \boldsymbol{w}(u)}\right]_{\boldsymbol{w}=\boldsymbol{w}_i} \quad (13)$$

where α_i is a positive scalar. On inserting (12) into (13) and using the constraint (4), we arrive at the following formula:

$$\boldsymbol{w}_{i+1}(u) = \boldsymbol{w}_i(u) + \alpha_i\boldsymbol{p}_i(u) \quad (14)$$

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

$$\boldsymbol{p}_i(u) = -\bar{P}\cdot\sum_v \bar{R}(u, v)\boldsymbol{w}_i(v) \quad (15)$$

and $\bar{P}$ is a $K\times K$ singular matrix of the form

$$\bar{P} = \left(\bar{I} - \frac{1}{K}11^T\right). \quad (16)$$

The vectors $\boldsymbol{w}(u)$ and $\boldsymbol{p}_i(u)$ 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:

$$E^N = \underbrace{E^K \times E^K \times \cdots \times E^K}_{L}. \quad (17)$$

Then the constraint (4) specifies a $(K-1)$-dimensional hyperplane in each E^K:

$$S_0: \boldsymbol{w}^T(u)\cdot 1 = 0, \qquad u \neq 0 \quad (18)$$

$$S_1: \left(\boldsymbol{w}(u) - \frac{1}{K}1\right)^T\cdot 1 = 0, \quad u = 0. \quad (19)$$

The hyperplane S_0 contains the origin of E^K; the hyperplane S_1 contains a vector $(1/K)$ 1 and is parallel to S_0. Then it is clear that $\bar{P}$ of (16) is the projection operator from E^K into S_0. Similarly for any point $\boldsymbol{x}$ in E^K, its projection into S_1 is given by $\bar{P}\boldsymbol{x}+(1/K)\cdot 1$.

As can be seen from the definition (15), the direction vector $\boldsymbol{p}_i(u)$ is the projection of the gradient $\Sigma_v\bar{R}(u, v)$

$w_i(v)$ into subspace Σ_0 which is a product of L copies of the hyperplane S_0:

$$\Sigma_0 = \underbrace{S_0 \times S_0 \times \cdots \times S_0}_{L}. \tag{20}$$

The dimension of Σ_0 is $N' = (K-1)L$. We define another N'-dimensional subspace Σ_1 by

$$\Sigma_1 = \underbrace{S_0 \times \cdots \times S_0}_{L_1} \times S_1 \times \underbrace{S_0 \times \cdots \times S_0}_{L_2}. \tag{21}$$

If the initial choice $\boldsymbol{w}_0(u)$ is in Σ_1, then $\boldsymbol{w}_i(u)$ also lies in the subspace Σ_1 for all i. We choose the gain α_i in such a way that the next approximation $\boldsymbol{w}_{i+1}(u)$ is the point which minimizes J of (12) over all points on the line of action of the vector $\boldsymbol{p}_i(u)$ passing through $\boldsymbol{w}_i(u)$. This leads us to the following recursive formula.

Initialization:

$$w_0 \in \Sigma_1 \tag{22a}$$

$$p_0 = -PRw_0. \tag{22b}$$

For $i \geqq 0$:

$$\alpha_i = \frac{|p_i|^2}{(p_i, Rp_i)} \tag{23a}$$

$$w_{i+1} = w_i + \alpha_i p_i \tag{23b}$$

$$p_{i+1} = PRw_{i+1} = p_i - \alpha_i PRp_i. \tag{23c}$$

In (22) and (23) we adopted the simplified notation R, p_i, w_i instead of $\overline{\boldsymbol{R}}(u, v)$, $\boldsymbol{p}_i(u)$, $\boldsymbol{w}_i(u)$, etc. Equation (22b) should read

$$\boldsymbol{p}_0(u) = -\overline{\boldsymbol{P}} \sum_v \overline{\boldsymbol{R}}(u, v)\boldsymbol{w}_0(v). \tag{24}$$

Similarly

$$|p_i|^2 = (p_i, p_i) = \sum_u \boldsymbol{p}_i^T(u)\boldsymbol{p}_i(u) \tag{25}$$

and

$$(p_i, Rp_i) = \sum_u \sum_v \boldsymbol{p}_i^T(u)\overline{\boldsymbol{R}}(u, v)\boldsymbol{p}_i(v). \tag{26}$$

From (23a) through (23c), it follows that the projected gradient vector $\boldsymbol{p}_{i+1}(u)$ is orthogonal to the previous one, $\boldsymbol{p}_i(u)$, i.e.,

$$(p_{i+1}, p_i) = \sum_u \boldsymbol{p}_{i+1}^T(u)\boldsymbol{p}_i(u) = 0. \tag{27}$$

The sequence of the ouput noise powers is monotone decreasing:

$$J_{i+1} - J_i = -\frac{|p_i|^4}{(p_i, Rp_i)} \leqq 0. \tag{28}$$

Therefore J_0 can be written as

$$J_0 = \sum_{i=1}^{\infty} \frac{|p_i|^4}{(p_i, Rp_i)}. \tag{29}$$

Since J_0 is finite, $|p_i|^4$ must converge to zero. Note that the null space of the projection operator $\overline{\boldsymbol{P}}$ consists of a set of vectors $c\,\mathbf{1}$ where c is a scalar constant. Hence from (15) the sequence $\{\boldsymbol{w}_i(u)\}$ converges to $\boldsymbol{w}_{\text{opt}}(u)$ of (7). Furthermore, it can be shown (Appendix I) that the sequence $\{J_i - J_\infty\}$ decreases by a factor of at least $\lambda_{\min}/\lambda_{\max}$ at each iteration, i.e.,

$$J_{i+1} - J_\infty \leqq \left(1 - \frac{\lambda_{\min}}{\lambda_{\max}}\right)(J_i - J_\infty) \tag{30}$$

where $\lambda_{\min}$ and $\lambda_{\max}$ are the maximum eigenvalue and the minimum positive eigenvalue of the covariance matrix function $\overline{\boldsymbol{\Phi}}(u, v) = \overline{\boldsymbol{P}} \cdot \overline{\boldsymbol{R}}(u, v) \cdot \overline{\boldsymbol{P}}$, respectively. From (30) it immediately follows that the convergence of $\{J_i\}$ is geometric:

$$J_i \leqq J_\infty + \rho^i(J_0 - J_\infty) \tag{31a}$$

where

$$\rho = 1 - \frac{\lambda_{\min}}{\lambda_{\max}} < 1. \tag{31b}$$

Although (22a) through (23c) appear to require computation of the cross-correlation function $\overline{\boldsymbol{R}}(u, v)$, it can be written in the following way by substituting the definition (6).

Initialization:

$$\boldsymbol{w}_0(u) \in \Sigma_1 \tag{32a}$$

$$\boldsymbol{p}_0(u) = -\langle \mathbf{x}(t-u) - x_{\text{av}}(t-u)\mathbf{1}, y_0(t)\rangle \tag{32b}$$

where

$$y_0(t) = \sum_u \boldsymbol{w}_0^T(u)\mathbf{x}(t-u) \tag{32c}$$

$$x_{\text{av}}(t) = \frac{1}{K}\mathbf{1}^T \cdot \mathbf{x}(t). \tag{32d}$$

For $i \geqq 0$:

$$q_i(t) = \sum_u \boldsymbol{p}_i^T(u)\mathbf{x}(t-u) \tag{33a}$$

$$\alpha_i = |p_i|^2 / \|q_i(t)\|^2 \tag{33b}$$

$$w_{i+1} = w_i + \alpha_i p_i \tag{33c}$$

$$\boldsymbol{p}_{i+1}(u) = \boldsymbol{p}_i(u) - \alpha_i\langle \mathbf{x}(t-u) - x_{\text{av}}(t-u)\mathbf{1}, q_i(t)\rangle \tag{33d}$$

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

$$\langle f(t), g(t)\rangle = \frac{1}{N_f}\sum_{t \in T_f} f(t)g(t). \tag{34}$$

One may replace (33b) and (33c) by the following:

$$\alpha_i = -\langle q_i(t), y_i(t)\rangle / \|q_i(t)\|^2 \tag{33b'}$$

and

$$y_{i+1}(t) = y_i(t) + \alpha_i q_i(t). \tag{33c'}$$

B. The Method of Conjugate Gradient with Projection

In the method of steepest descent, the projected gradient 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 $\boldsymbol{w}(u)$ is finite.

In the present section we will modify the fundamental conjugate gradient method (Appendix II) so as to be able to apply the method to our specific problem. The method of conjugate gradients was devised by Hestenes and Stiefel [6] to solve a system of simultaneous linear algebraic equations,

$$\bar{\boldsymbol{A}}\boldsymbol{x} = \boldsymbol{b} \tag{35}$$

where where $\bar{\boldsymbol{A}}$ is an $N\times N$ positive definite matrix, $\boldsymbol{x}$ an $N\times 1$ vector of unknowns, and $\boldsymbol{b}$ is an $N\times 1$ 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 $M\leqq N$. Clearly the same algorithm can be applied to find $\boldsymbol{x}$ which minimizes the following function:

$$f(\boldsymbol{x}) = \tfrac{1}{2}\boldsymbol{x}^T\bar{\boldsymbol{A}}\boldsymbol{x} - \boldsymbol{x}^T\boldsymbol{b}. \tag{36}$$

By modifying the fundamental formula we can obtain the following conjugate gradient iterative procedure leading to the minimization of the quadratic form (5) under the fidelity constraint.

Initialization:

$$w_0 \in \Sigma_1 \tag{37a}$$

$$p_0 = r_0 = -PRw_0. \tag{37b}$$

For $i\geqq 0$:

$$\alpha_i = \|r_i\|^2/(p_i, Rp_i) \tag{38a}$$

$$w_{i+1} = w_i - \alpha_i p_i \tag{38b}$$

$$r_{i+1} = -PRw_{i+1} = r_i - \alpha_i PRp_i \tag{38c}$$

$$\beta_i = |r_{i+1}|^2/|r_i|^2 \tag{38d}$$

$$p_{i+1} = r_{i+1} + \beta_i p_i. \tag{38e}$$

In place of (38a) and (38d) one may use

$$\alpha_i = (p_i, r_i)/(p_i, Rp_i) \tag{38a'}$$

$$\beta_i = -(r_{i+1}, Rp_i)/(p_i, Rp_i) \tag{38d'}$$

where P is the projection operator defined by the matrix (16).

Many relations hold among the quantities appearing in (37a) through (38e). The most important ones are

$$p_i \in \Sigma_0, \quad r_i \in \Sigma_0, \quad w_i \in \Sigma_1, \qquad \text{for all } i \tag{39}$$

$$(r_i, r_j) = 0, \qquad i \neq j \tag{40}$$

$$(p_i, Rp_j) = 0, \qquad i \neq j. \tag{41}$$

The projected gradients $\{r_i;\ i=0, \cdots, N'-1\}$ where $N'=(K-1)L$ form a set of orthogonal vectors in the subspace Σ_0. The direction vectors $\{p_i; i=0, \cdots, N'-1\}$ form a set of R-conjugate or R-orthogonal vectors [6], [7] 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 $\{p_j\}$. In fact, using the coefficients α_1 of (38a), we have the relation

$$w_{\text{opt}} - w_0 = \sum_{i=0}^{N'-1} \alpha_i p_i \in \Sigma_0. \tag{42}$$

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

$$\begin{aligned} J_{i-1} - J_i &= -(p_i, r_i)^2/(p_i, Rp_i) \\ &= -|r_i|^4/(p_i, Rp_i) \leqq 0. \end{aligned} \tag{43}$$

Therefore, the sequence $\{r_i\}$ must converge to zero and the convergence of $\{w_i(u)\}$ to $w_{\text{opt}}(u)$ can be shown from the relation (38c). Furthermore, $w_j(u)$ is closer to the solution $w_{\text{opt}}(u)$ than $w_i(u)$, $i\leqq j$ (see Appendix II). The result indicates that if we stop the iterative process at any step, the last obtained approximation is the best in the sense of being the closest to the true solution. An upper bound for the rate of convergence can be obtained as in the case of the steepest descent method (Appendix III):

$$J_i \leqq J_\infty + \rho^i(J_0 - J_\infty). \tag{44}$$

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

Initialization:

$$w_0 \in \Sigma_1 \tag{45a}$$

$$p_0(u) = r_0(u) = -\langle \mathbf{x}(t-u) - x_{\text{av}}(t-u)\mathbf{1}, y_0(t)\rangle. \tag{45b}$$

For $i\geqq 0$:

$$\alpha_i = |r_i|^2/\|q_i(t)\|^2 \tag{46a}$$

$$w_{i+1} = w_i + \alpha_i p_i \tag{46b}$$

$$r_{i+1}(u) = r_i(u) - \alpha_i\langle \mathbf{x}(t-u) - x_{\text{av}}(t-u)\mathbf{1}, q_i(t)\rangle \tag{46c}$$

$$\beta_i = |r_{i+1}|^2/|r_i|^2 \tag{46d}$$

$$p_{i+1} = r_{i+1} + \beta_i p_i \tag{46e}$$

where the functions $y_0(t)$, $x_{\text{av}}(t)$, and $q_i(t)$ are defined by (32c), (32d), and (33a).

Equations (46a) and (46b) can be replaced by

$$\alpha_i = -\langle q_i(t), y_i(t)\rangle/\|q_i(t)\|^2 \tag{46a'}$$

and

$$y_{i+1}(t) = y_i(t) + \alpha_i q_i(t). \tag{46b'}$$

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 for 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 previously.

A. The Steepest Descent (SD) Method

The input data $\mathbf{x}(t)$ must be stored in any method and, therefore, we exclude the space for $\mathbf{x}(t)$ 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. 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 allotted to $y(t)$. In this respect the minimum memory requirement is

$$M_{SD} \cong (N_f + 2KL) \quad \text{words.} \tag{47}$$

As to the computation time, the majority is spent for the convolutional sum to obtain $q_i(t)$ and $\boldsymbol{p}_i(u)$ 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,

$$T_{SD} = 2KLN_f(\mu + \nu) \quad \text{seconds/iteration} \tag{48}$$

is the running time.

B. The Conjugate Gradient (CG) Method

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

$$M_{CG} \cong (N_f + 3KL) \quad \text{words.} \tag{49}$$

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 $\boldsymbol{r}_i(u)$. Therefore,

$$T_{CG} = (2KLN_f + 3KL)(\mu + \nu) \tag{50}$$

is the running time.

IV. Comparison with Other Design Methods

In this section we will compare the performance of the iterative procedures discussed in the preceding sections with the other synthesis methods: the time-domain design method and the frequency-domain design method.

A. The Time-Domain (TD) Design Method

The most straightforward method for synthesizing the filter $\boldsymbol{w}(u)$ is the time-domain method in which the cross-correlation function $\bar{\boldsymbol{R}}(u, v)$ is actually calculated from the input data in the fitting interval T_f, and (7) and (8) are then used. The major portion of the memory requirements are those for the correlation function which takes K^2L^2 words (excluding the working space for matrix inversion):

$$M_{TD} \cong K^2L^2 \text{ words} \tag{51}$$

which may well exceed M_{SD} and M_{CG} when either K or L is large.

As to the computation time, $\bar{\boldsymbol{R}}(u, v)$ and its inversion take about $K^2L^2N_f$ and $(KL)^3/3$ operations, respectively. Therefore, the total computation time is

$$T_{TD} \cong (K^2L^2N + \tfrac{1}{3}K^3L^3)(\mu + \nu) \quad \text{seconds.} \tag{52}$$

Here we assume that the standard matrix inversion subroutine with Gauss elimination method is used. If N_f is large enough, i.e., $N_f \gg L$, then it follows that

$$\bar{\boldsymbol{R}}(u, v) \cong \bar{\boldsymbol{R}}(u - v). \tag{53}$$

In this case the efficient recursive formula for inversion of Toepolitz matrices devised by Wiggins and Robinson [9] can be applied. The computation time is approximately given by

$$T_{TD}' \cong (K^2LN_f + 2.5L^2K^3)(\mu + \nu) \quad \text{seconds.} \tag{54}$$

B. The Frequency-Domain (FD) Synthesis Method

A method for obtaining $\boldsymbol{w}_{opt}(u)$ approximately in the frequency domain has been discussed extensively by Capon *et al.* [2].

Relabeling the argument of $\boldsymbol{w}(u)$ from 0 to $L-1$ instead of $-L_1$ to L_2, and applying the finite Fourier transform, P_{out} of (5) can be written in the form

$$\begin{aligned} P_{out} &= \sum_u \sum_v \boldsymbol{w}^T(u)\bar{\boldsymbol{R}}(u, v)\boldsymbol{w}(v) \\ &= L \sum_{f_1} \sum_{f_2} \boldsymbol{A}(f_1)\bar{\boldsymbol{P}}(f_1, f_2)\boldsymbol{A}(f_2) \end{aligned} \tag{55}$$

where

$$A(f) = \frac{1}{L} \sum_u \boldsymbol{w}(u) \exp(-j2\pi fu/L) \tag{56}$$

and

$$\begin{aligned} \bar{\boldsymbol{P}}(f_1, f_2) = \frac{1}{L} \sum_u \sum_v \bar{\boldsymbol{R}}(u, v) \\ \cdot \exp\{-j2\pi(f_1u - f_2v)/L\}. \end{aligned} \tag{57}$$

The two-dimensional finite Fourier transform $\bar{\boldsymbol{P}}(f_1, f_2)$ can be reduced to the form

$$\bar{\boldsymbol{P}}(f_1, f_2) = \bar{\boldsymbol{P}}(f_1)\delta(f_1 - f_2) \tag{58}$$

if and only if

$$\bar{\boldsymbol{R}}(u, v) = \bar{\boldsymbol{R}}(u - v) \tag{59}$$

and

$$\bar{\boldsymbol{R}}(u - L) = \bar{\boldsymbol{R}}(u), \qquad 0 \leqq u \leqq L - 1. \tag{60}$$

The condition (59) is practically satisfied, as was mentioned earlier, but (60) is far from the reality since the input $\mathbf{x}(t)$ need be periodic with period L in order for the condition to hold. However, if the filter length L, is

so large that the correlation function satisfies

$$\bar{R}(u) \cong 0, \quad u > L/2 \tag{61}$$

then the corresponding solution $w_{opt}(u)$ will satisfy $w_{opt}(u) \cong 0$, for $u > L/2$. Then the quadratic form (5) will remain the same even if $\bar{R}(u)$ is replaced by $\bar{R}(u-L)$, for $L-1 \geqq u > L/2$.

If we accept the approximations (59) and (60), the power spectrum $\bar{P}(f)$ is given by

$$\bar{P}(f) = \frac{1}{LN_f} \sum_{t \in T_f} \tilde{X}(f; t) X^T(f; t), \qquad f = 0, 1, \cdots, L' - 1 \tag{62}$$

where $\sim$ means complex conjugate and where

$$X(f; t) = \sum_{u=0}^{L-1} x(t + u - L + 1) \exp \{-j2\pi fu/L\}. \tag{63}$$

Here $L' = (L+1)/2$ for L odd, and $(L/2)+1$ for L even. After the cross-power spectrum is obtained, the filter response function is given by

$$w(u) = \frac{1}{L} \sum_{f=0}^{L'-1} \frac{\bar{P}^{-1}(f)1}{1^T \bar{P}^{-1}(f)1} \exp \{j2\pi f(u - L_1 - 1)/L\}. \tag{64}$$

The majority of the memory requirement in this method are for $\bar{P}(f)$ which takes about $(2K)^2 L'$:

$$M_{FD} \cong 2K^2 L \text{ words.} \tag{65}$$

The number of the basic operations for the calculation of $X(f; t)$ and for the cross-power spectrum are approximately KL^2N_f and $K(K+1)\ LN_f$, respectively. The matrix inversion takes $4K^3L/3$ operations by the Gauss elimination method. Hence an estimate for the total computation time is

$$T_{FD} \cong \{KL(K+L)N_f + 4K^3L/3\}(\mu + \nu) \text{ seconds.} \tag{66}$$

The direct segment method [2] for the spectral matrix calculation is equivalent to approximating $\bar{P}(f)$ by

$$\bar{P}(f) \cong \frac{1}{LM} \sum_{n=0}^{M-1} \tilde{X}(f; nD) X^T(f; nD) \tag{67}$$

where M is the number of segments in the fitting interval and D is the distance between the first data points of the neighboring segments, i.e., $M = N_f/D$. Use of the segmentation method is equivalent, in the time domain, to replacing the criterion P_{out} of (5) by

$$P'_{out} = \frac{1}{M} \sum_{n=0}^{M-1} y^2(nD) \tag{68}$$

which is the output noise power averaged over D other points in the fitting interval.

It should be noted that the reduction of computing time by use of segmentation is not a feature inherent in the frequency-domain synthesis procedure, but is simply the result of the replacement of a design criterion P_{out} by P'_{out}. Obviously, the segmentation method can be equally applicable to the iterative methods of Section II and to the time-domain synthesis procedure as well, and it will cut the computation time, roughly, by a factor of D.

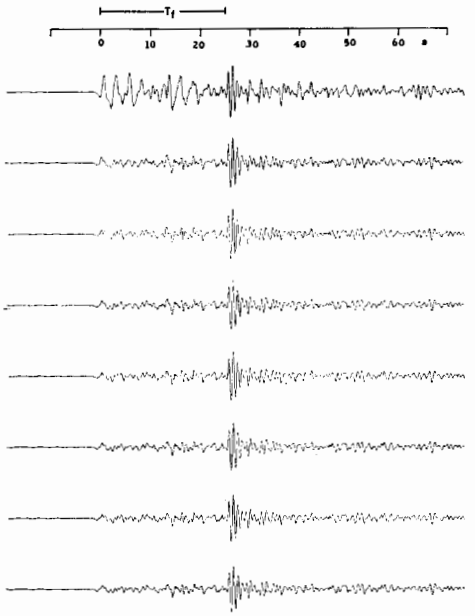

Fig. 1. Processed outputs by steepest descent method, $L_1 = L_2 = 0$, $T_f = 25$ seconds, $s(t) = 0$ for $t \in T_f$.

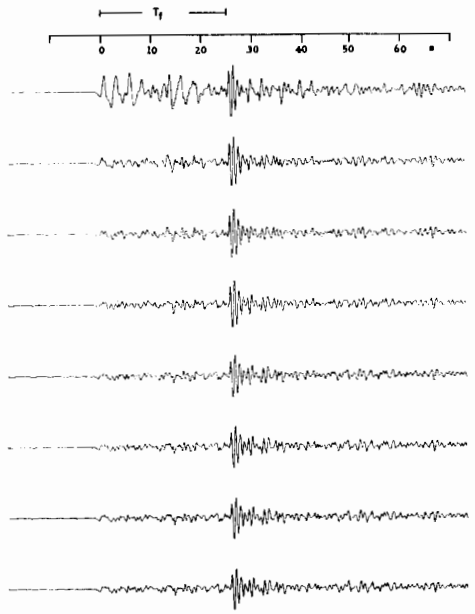

Fig. 2. Processed outputs by conjugate gradient method, $L_1 = L_2 = 0$, $T_f = 25$ seconds, $s(t) = 0$ for $t \in T_f$.

V. Simulation Results

The iterative procedures outlined above were applied to the data from the Montana large aperture seismic array. The 21 center seismometers' outputs were used,

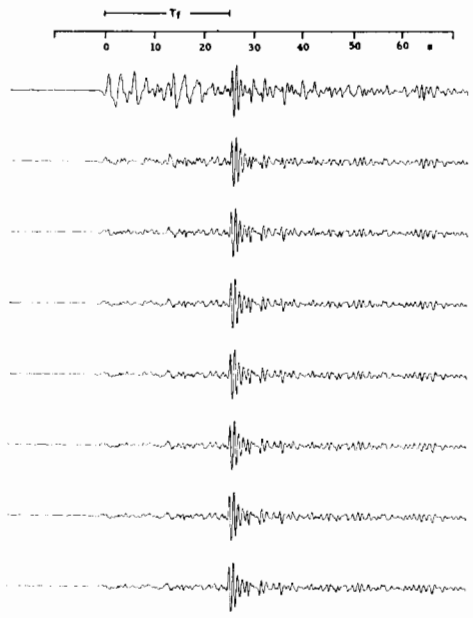

Fig. 3. Processed outputs by steepest descent method, $L_1 = L_2 = 10$, $T_f = 25$ seconds, $s(t) = 0$ for $t \in T_f$.

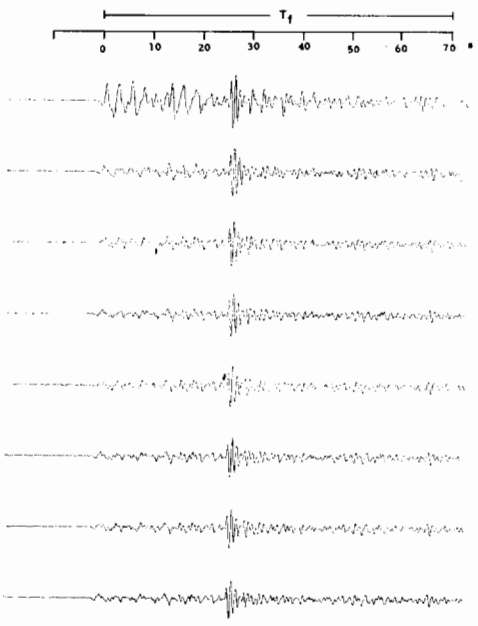

Fig. 5. Processed outputs by the steepest descent method, $L_1 = L_2 = 0$, $T_f = 70$ seconds, $s(t) \neq 0$ for $t \in T_f$.

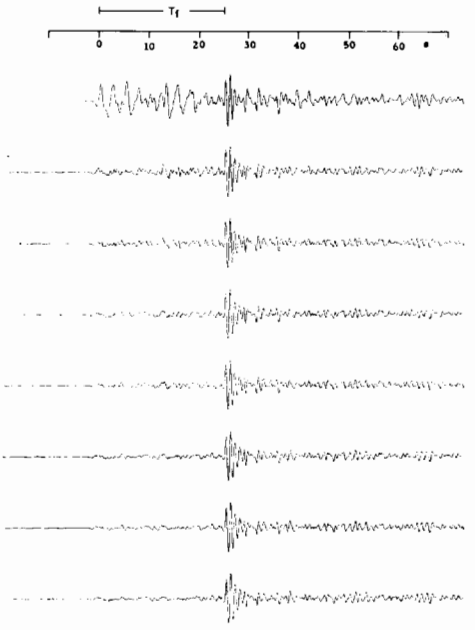

Fig. 4. Processed outputs by the conjugate gradient method, $L_1 = L_2 = 10$, $T_f = 25$ seconds, $s(t) = 0$ for $t \in T_f$.

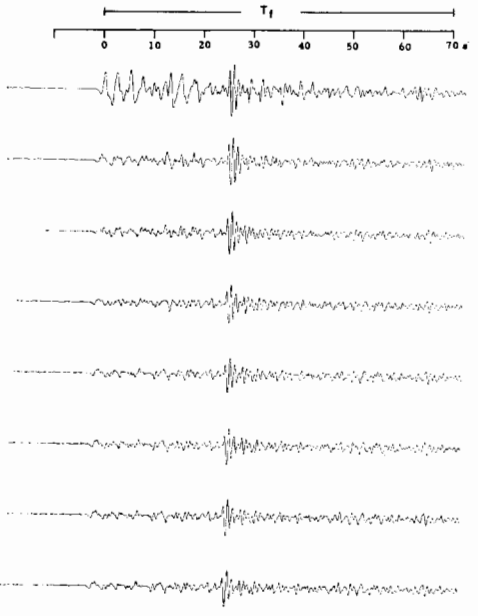

Fig. 6. Processed outputs by the conjugate gradient method $L_1 = L_2 = 0$, $T_f = 70$ seconds, $s(t) \neq 0$ for $t \in T_f$.

i.e., $K = 21$. The Longshot explosion's data (October 29, 1965) were superposed on the Kamchatka earthquake data (April 8, 1966) with a 25-second delay.

Fig. 1 shows processed outputs $y_i(t)$ of the first eight iteration procedures by use of the steepest descent method, where $L_1 = L_2 = 0$; i.e., a simple multiply-and-sum filter. The fitting interval T_f is the first 25 seconds after the arrival of the Kamchatka earthquake, and $N_f = 500$ since the data sampling rate is 20 Hz. Note that the end of T_f is just before the arrival of the second event. The iteration starts from the simple beam forming, i.e., $\boldsymbol{w}_0(0) = (1/K)\mathbf{1}$. The top curve of Fig. 1 is $y_0(t) = x_{av}(t)$. We notice that the interfering event (i.e., the first event) is suppressed drastically (about 9 dB) after the first iteration and that an approximate solution attained after a few iterations virtually satisfies the practical purpose. Fig. 2 is the result obtained by the conjugate gradient method where $L_1 = L_2 = 0$, and $\boldsymbol{w}_0(0) = (1/K)\mathbf{1}$. The first two steps, i.e., $y_0(t)$ and $y_1(t)$, are, by definition, the same as those that are shown in Fig. 1, but some improvement can be noticed in the following steps.

Figs. 3 and 4 are the results obtained by the steepest descent method and the conjugate gradient method, respectively, when 10 points in the past and 10 points in the future are used, i.e., $L_1 = L_2 = 10$. The iteration starts from a simple sum, i.e., $w_0(u) = (1/K)\delta_{u,0}\mathbf{1}$. The interfering event is suppressed 11 dB by the first iteration, and again, a few iterations seem to be enough for practical purposes.

Figs. 5 and 6 show processed outputs of the steepest descent method for $L_1 = L_2 = 0$ when the second event $s(t)$ is present in the fitting interval T_f, which is chosen as long as 70 seconds after the first event arrival. Initial setting is again $w_0(u) = (1/K)\delta_{u,0}\mathbf{1}$. Here some decrease in the amplitude of the signal is observed. This effect can be ascribed to the error of the approximation (10) and to the discrepancy between the actual signal and the idealized signal model (1) in which we assume signal components are common to all seismometers.

Fig. 7 is a plot of the output noise power versus 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 (CG) method shows a faster convergence than the steepest descent (SD) method. We also synthesized filters in the frequency domain [2], where filter lengths of $L = 8$ and 16 were chosen so that the fast Fourier transform subroutine could be utilized. Reduction of the noise power output attained by those filters is also indicated in Fig. 7, 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 as follows.

	$L=1$ (seconds)	$L=11$ (seconds)	$L=21$ (seconds)
SD method	4.28	38.16	72.30
CG method	5.07	46.00	87.27

On the other hand, the frequency-domain synthesis takes 622.0 seconds for $L = 8$ and 1135 seconds for $L = 16$, where $D = 1$ was used in (67). 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 provides an efficient way of synthesizing array processors.

Appendix I
The Rate of Convergence of the Steepest Descent Method

In this Appendix and in Appendix III we adopt the simplified notation as in Section II [see (24) through (26)].

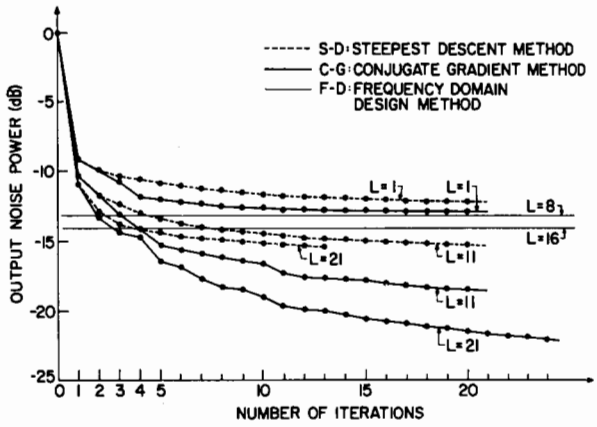

Fig. 7. Output noise power versus number of iterations.

Since $\{w_i\}$ converges to w_{opt} of (7), the output noise sequence $\{J_i\}$ converges to the following value:

$$J_\infty = (w_{opt}, Rw_{opt}). \tag{69}$$

Let us change variables from w_i to f_i by the following relation

$$f_i = w_i - w_{opt}, \qquad f_i \in \Sigma_0. \tag{70}$$

Then J_i can be written into the form

$$\begin{aligned} J_i &= (f_i + w_{opt},\ R(f_i + w_{opt})) \\ &= (f_i, Rf_i) + J_\infty = K_i + J_\infty. \end{aligned} \tag{71}$$

Now we rewrite the numerator of (28) using (23b) and (70)

$$|p_i|^2 = \|PRw_i\|^2 = (f_i, RPRf_i) \tag{72}$$

where we used the relation

$$PRw_{opt} = 0. \tag{73}$$

Since the vector f_i is constrained to the subspace Σ_0 it is convenient to write f_i as

$$f_i = P \cdot g_i \tag{74}$$

with no constraint on g_i. Then (72) can be written as follows:

$$|p_i|^2 = (g_i, \Phi^2 g_i) \tag{75}$$

where

$$\Phi = P \cdot R \cdot P. \tag{76}$$

Similarly the denominator of (28) can be written as follows:

$$(p_i, Rp_i) = (g_i, \Phi^3 g_i). \tag{77}$$

The quantity K_i defined by (71) can be likewise written as

$$K_i = (f_i, Rf_i) = (g_i, \Phi g_i). \tag{78}$$

Therefore from (28) and (75)–(78), it follows that

$$K_{i+1} = K_i \left\{1 - \frac{(g_i, \Phi^2 g_i)}{(g_i, \Phi g_i)} \cdot \frac{(g_i, \Phi^2 g_i)}{(g_i, \Phi^3 g_i)}\right\}. \tag{79}$$

The explicit representation for Φ of (76) is

$$\bar{\Phi}(u, v) = \bar{P} \cdot \bar{R}(u, v) \cdot \bar{P} \tag{80}$$

which represents the covariance function of the projected process

$$\bar{P} \cdot \mathbf{x}(t) = \mathbf{x}(t) - x_{\mathrm{av}}(t) \cdot \mathbf{1}. \tag{81}$$

Clearly $\bar{\Phi}(u, v)$ is a nonnegative definite and its $N(=KL)$ eigenvalues are all positive except for one which is $\lambda_0 = 0$. Letting $\lambda_{\max}$ be the largest eigenvalue and $\lambda_{\min}$ be the smallest eigenvalue, excluding λ_0, of the covariance function Φ, then clearly,

$$\lambda_{\max} \geqq \lambda_{\min} > \lambda_0 = 0. \tag{82}$$

Any given vector $g_i \in E^N$ can be uniquely decomposed into the two components $g_i^{(1)}$ and $g_i^{(2)}$:

$$g_i^{(1)} = P \cdot g_i = f_i \in \Sigma_0 \tag{83}$$

and

$$g_i^{(2)} = g_i - P \cdot g_i \perp \Sigma_0. \tag{84}$$

If $g_i^{(1)} = f_i = 0$, then $w_i = w_{\mathrm{opt}}$ and the solution is attained. Therefore, we assume that $f_i \neq 0$, and the following inequalities hold for all i:

$$\frac{(g_i, \Phi^2 g_i)}{(g_i, \Phi g_i)} = \frac{(f_i, \Phi^2 f_i)}{(f_i, \Phi f_i)} \geqq \lambda_{\min} \tag{85}$$

and

$$\frac{(g_i, \Phi^2 g_i)}{(g_i, \Phi^3 g_i)} = \frac{(f_i, \Phi^2 f_i)}{(f_i, \Phi^3 f_i)} \geqq \frac{1}{\lambda_{\max}}. \tag{86}$$

Then (79), (85), and (86) lead to (30) immediately.

Appendix II
Basic Algorithm of the Conjugate Gradient Method

Let x_0 be an arbitrary starting approximation to the solution vector of (35). Then, the following formulas define the fundamental conjugate gradient iterative procedure leading to the solution $h = A^{-1}b$ [6]–[8]:

$$p_0 = r_0 = b - Ax_0 \tag{87}$$

$$\alpha_i = |r_i|^2/(p_i, Ap_i) \tag{88a}$$

$$x_{i+1} = x_i + \alpha_i p_i \tag{88b}$$

$$x_{i+1} = b - Ax_{i+1} = r_i - \alpha_i A p_i \tag{88c}$$

$$\beta_i = |r_{i+1}|^2/|r_i|^2 \tag{88d}$$

$$p_{i+1} = r_{i+1} + \beta_i p_i. \tag{88e}$$

After M iterations, with $M \leqq N$, x_M will be equal to the solution h if all computations are done with no loss of accuracy. Many relations hold among the quantities appearing in (88a) through (88e):

$$(r_i, r_j) = 0, \qquad i \neq j \tag{89a}$$

$$(p_i, Ap_j) = 0, \qquad i \neq j \tag{89b}$$

$$(p_i, r_j) = 0, \qquad i < j \tag{89c}$$

$$(p_i, r_j) = |r_i|^2, \qquad i \geqq j \tag{89d}$$

$$(r_i, Ap_i) = (p_i, Ap_i) \tag{89e}$$

$$(r_i, Ap_j) = 0, \qquad i \neq j, j+1. \tag{89f}$$

Using the properties (89d), the sequences α_i and β_i satisfy

$$\alpha_i = \frac{|r_i|^2}{(p_i, Ap_i)} = \frac{(p_i, r_i)}{(p_i, Ap_i)} \tag{90a}$$

$$\beta_i = \frac{|r_{i+1}|^2}{|r_i|^2} = -\frac{(r_{i+1}, Ap_i)}{(p_i, Ap_i)}. \tag{90b}$$

Some other salient relations that hold among the quantities appearing in the iterative formulas are

$$|x_i - h| > |x_j - h|, \quad i < j \tag{91}$$

$$(x_i - h, A^{-1}(x_i - h)) > (x_j - h, A^{-1}(x_j - h)), \quad i < j \tag{92}$$

and

$$\frac{|p_i|^2}{(p_i, Ap_i)} > \frac{|r_i|^2}{(p_i, Ap_i)} > \frac{|r_i|^2}{(r_i, Ar_i)}. \tag{93}$$

Equation (92) will be used in Appendix III where we discuss the rate of convergence of the conjugate gradient method.

Appendix III
The Rate of Convergence of the Conjugate Gradient Method

The argument required for a proof of the inequality (44) is essentially the same as in Appendix I. The only difference is the right-hand sides of (28) and of (43). However, using the inequality relation (93) we have the following:

$$\frac{|r_i|^4}{(p_i, Rp_i)} > \frac{|r_i|^4}{(r_i, Rr_i)} = \frac{(g_i, \Phi^2 g_i)}{(g_i, \Phi^3 g_i)} \tag{94}$$

where we used the relation

$$r_i = -PRw_i = -PRf_i = -\Phi g_i \tag{95}$$

where f_i, g_i, and Φ are defined by (70), (74), and (76), respectively. The rest of the proof is exactly the same as in Appendix I.

Acknowledgment

The author is indebted to Dr. P. D. Welch, IBM Research Center, for the helpful suggestions and criticism he provided, and also, wishes to thank G. H. Purdy for his programming support.

References

[1] E. J. Kelly and M. J. Levin, "Signal parameter estimation for seismometer arrays," M.I.T. Lincoln Lab., Lexington, Mass., Tech. Rept. 339, April 1964.
[2] J. Capon, R. J. Greenfield, and R. J. Kolker, "Multidimensional maximum-likelihood processing of a large aperture seismic array," *Proc. IEEE*, vol. 55, pp. 192–211, February 1967.
[3] A. V. Balakrishnan, "Filtering and prediction theory," in *Communication Theory*, A. V. Balakrishnan, Ed. New York: McGraw-Hill, 1968.
[4] B. Widrow, P. E. Mantey, L. J. Griffiths, and B. B. Goode, "Adaptive antenna systems," *Proc. IEEE*, vol. 55, pp. 2143–2159, December 1967.
[5] R. T. Lacoss, "Adaptive combining of wideband array data for optimal reception," *IEEE Trans. on Geoscience Electronics*, vol. GE-6, pp. 78–86, May 1968.
[6] M. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *J. Res. Natl. Bur. Std.*, vol. 49, pp. 409–436, December 1952.
[7] F. S. Beckman, "The solution of linear equations by the conjugate gradient method," in *Mathematical Methods for Digital Computers*, A. Ralston and H. S. Wilf, Eds. New York: Wiley, 1960.
[8] R. Fletcher and M. J. D. Powell, "A rapidly convergent descent method for minimization," *Computer J.*, vol. 6, pp. 163–168, 1963.
[9] R. A. Wiggins and E. A. Robinson, "Recursive solution to the multichannel filtering problem," *J. Geophys. Res.*, vol. 70, pp. 1885–1891, April 15, 1965.
[10] H. Kobayashi and P. D. Welch, "The recursive design of a seismic array process," presented at the 3rd Ann. Princeton Conf. Information Science and Systems, March 27–28, 1969.