

# Implementation of Adaptive Array Algorithms

ROBERT SCHREIBER

**Abstract**—Some new, efficient, and numerically stable algorithms for the recursive solution of matrix problems arising in optimal beamforming and direction finding are described and analyzed. The matrix problems considered are systems of linear equations and spectral decomposition. While recursive solution procedures based on the matrix inversion lemma may be unstable, ours are stable. Furthermore, these algorithms are extremely fast.

## I. INTRODUCTION

IN this paper we consider the computational procedures to be used in implementing some standard and some more recently proposed adaptive methods for direction finding and beamforming by sensor arrays. We discuss the computation of a minimum variance distortionless response (MVDR) beamformer and of several high-resolution methods (recently advocated by Bienvenue and Mermoz [1], Owsley [9], and Schmidt [10]) that are based on the spectral decomposition of the signal covariance matrix. We are especially concerned with recursive implementation of these procedures. Whenever the signal is sampled, an estimate for the covariance matrix is updated and the computed solution (a weight vector) changes in response to this new information. We shall propose and analyze some new, efficient, numerically stable algorithms.

The computational procedures we advocate take advantage of this on-line character. We find methods for updating the solutions that are much less expensive than procedures that do not make use of the previously computed solution.

For the MVDR method, some previous work has been done [8]. An update method based on the Sherman-Morrison-Woodbury formula (which is also known as the matrix inversion lemma) has been advocated. We show that this procedure can, in one common circumstance, be numerically unstable. We propose three new, stable methods here. For the high-resolution methods, we illustrate the use of some efficient procedures for updating eigenvalue and singular value decompositions. We show how to take advantage of the existence of multiple eigenvalues of the signal covariance matrix to further reduce the work. We also show that complex arithmetic can largely be avoided.

Manuscript received June 26, 1984; revised December 18, 1985. This work was supported by the U.S. Office of Naval Research under Contract N00014-82-K-0703 and by the U.S. Army Research Office under Contract DAAG 29-83-K-0124.

The author is with the Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY 12180. He is also a Consultant to Saxpy Computer Corporation, Sunnyvale, CA 94086.

IEEE Log Number 8609623.

## A. Notation

Let  $\mathbb{C}^n$  and  $\mathbb{C}^{m \times n}$  denote the spaces of complex  $n$  vectors and  $m \times n$  matrices. We use upper case italic letters for matrices, lower case italic letters for vectors. The  $m \times n$  matrix  $A$  has, by convention, the columns  $[a_1, a_2, \dots, a_n]$ , and the elements  $[\alpha_{i,j}]$ ; the vector  $x$  has elements  $(\xi_1, \dots, \xi_n)^T$ ; for  $A \in \mathbb{C}^{m \times n}$ ,  $A^T$  denotes the transpose and  $A^H$  the conjugate transpose of  $A$ . If  $A \in \mathbb{C}^{n \times n}$  is diagonal ( $\alpha_{i,j} = 0$  for  $i \neq j$ ), we denote  $A$  by  $\text{diag}(\alpha_{1,1}, \dots, \alpha_{n,n})$ . We denote the  $r \times r$  identity matrix by  $I_r$ . For  $A \in \mathbb{C}^{n \times n}$ , the Frobenius norm of  $A$  is given by

$$\|A\|_F \equiv \left( \sum_{i,j=1}^n |\alpha_{i,j}|^2 \right)^{1/2}.$$

In giving operation counts for algorithms, we use the term *operation* to mean one complex multiplication and one complex addition. One operation costs about as much as four real multiplications and four real additions. Note that computing  $x + \alpha y$  with real  $\alpha$  costs one-half an operation.

## II. AN ON-LINE ALGORITHM FOR ADAPTIVE BEAMFORMING

Let  $x \in \mathbb{C}^n$  be a narrow-band signal received by an array of  $n$  elements. Let its covariance matrix be denoted  $R$ ,

$$R \equiv E\{xx^H\}, \quad (1)$$

where  $E\{ \}$  denotes expected value.  $R$  is Hermitian and, if any noise is present, positive definite. Thus,  $R$  has a Cholesky factorization

$$R = LL^H \quad (2)$$

where  $L$  is lower triangular and has positive, real diagonal elements. The factorization can be computed in  $n^3/6$  operations [12]. With the help of the Cholesky factorization, we can compute  $R^{-1}d$  with  $n^2$  operations by solving two triangular systems:

$$Lu = d,$$

and

$$L^H w = u.$$

Thus,  $w = L^{-1}u = (LL^H)^{-1}d = R^{-1}d$ .

Consider the adaptive control of an  $n$ -element array. The minimum variance distortionless response beamformer determines the output of the array by

$$g(d) = w^H x \quad (3)$$

where  $g$  is an estimate of the signal arriving from some given bearing,  $d$  is a steering vector for the given array and bearing,  $x$  is the signal vector, and

$$w = R^{-1}d\rho(d), \tag{4}$$

where  $\rho(d)$  is an estimate of the average power arriving from the given bearing,

$$\rho(d) \equiv (d^H R^{-1} d)^{-1}. \tag{5}$$

In practice, we have several bearing angles and corresponding steering vectors  $d_i, i = 1, 2, \dots, m$ . Let  $D$  be the  $n \times m$  matrix of these vectors

$$D \equiv [d_1, \dots, d_m].$$

The principal computational problem is, then, to find the  $n \times m$  solution matrix

$$W \equiv R^{-1}D. \tag{6}$$

In the on-line beamforming problem,  $D$  remains fixed, but  $R$  is often changed to incorporate a new sample  $x$  of the signal:

$$*R = \mu R + (1 - \mu)xx^H \tag{7}$$

where  $\mu \in (0, 1)$ . We refer to such a change as a rank-one update to  $R$  (since the rank of  $xx^H$  is one) although, if  $\mu \neq 1$ , the change to  $R$  is, in general, of full rank. One must find the corresponding updated solution

$$*W = *R^{-1}D. \tag{8}$$

The obvious method [8] is based on the Sherman-Morrison-Woodbury formula for  $*R^{-1}$ :

$$*R^{-1} = \mu^{-1}R^{-1} + \beta zz^H \tag{9}$$

where

$$\beta = (\mu - 1)\mu^{-1}[(1 - \mu)(x^H R^{-1} x) + \mu]^{-1} \tag{10}$$

and

$$z = R^{-1}x. \tag{11}$$

Thus, if  $d$  is a typical column of  $D$ , and  $w$  is the corresponding column of  $W$ , then by (6), (8), and (9),

$$*w = \mu^{-1}w + \beta zz^H d. \tag{12}$$

To make use of the method (12) one requires that  $R^{-1}x$  be computed. This can be done with the aid of the Cholesky factorization of  $R$ . Moreover, Gill, Golub, Murray, and Saunders have suggested a method for updating the Cholesky factorization after the rank-one change (7) that uses about  $(\frac{3}{2})n^2 + O(n)$  operations [5]. Fast computation of this algorithm by parallel processor arrays was considered by Schreiber and Tang [11].

This suggests the following algorithm.

- 1) (Initialize.) Let  $R = l, L = l$ , and  $W = D$ . Thus,  $R = LL^H$ , and  $W = R^{-1}D$ . Compute  $\beta$  from (10).
- 2) Every time an update (7) is made to  $R$ ,
  - a) solve for  $z = R^{-1}x$  by solving the two triangular systems  $Ly = x$  and  $L^H z = y$ ;
  - b) update the Cholesky factor  $L$  of  $R$ ; and

c) for every column  $w$  of  $W$  and corresponding column  $d$  of  $D$ ,

- i) compute  $\delta := \beta z^H d$ ;
- ii) compute  $w := \mu^{-1}w + \delta z$ .

The cost of step 2a) is  $n^2$  operations; of step 2b) is  $(\frac{3}{2})n^2$  operations; of step 2ci) is  $nm$  operations; of step 2cii) is about  $(\frac{3}{2})nm$  operations. The alternative algorithm, in which the updated Cholesky factorization of  $R$  is used to solve for  $R^{-1}D$ , costs  $n^2 m$  operations for solving triangular systems and  $(\frac{3}{2})n^2$  for updating the factorization.

Unfortunately, the method is unstable. If  $0 < \mu < 1$ , then  $\mu^{-1} > 1$ . Any error in  $W$  is amplified by the factor  $\mu^{-1}$  every time the update (12) is performed. These errors eventually render the computed solutions  $W$  uselessly inaccurate. Thus, correct solutions must occasionally be calculated directly from  $D$  and the Cholesky factorization of  $R$  according to (8). Fortunately, in some applications one may take  $\mu \approx 1$  (so that the estimate of  $R$  computed using (7) is a better approximation to the true covariance matrix.) Thus,  $\mu^{-1}$  is only slightly larger than 1, and the "unstable" update (12) can be used for quite some time.

In fact, the choice  $\mu \approx 1$  is appropriate for relatively stationary signal environments. And in this case, it may also be allowable to avoid updating the weights with every new sample. But, in a rapidly changing environment, one would take  $\mu$  substantially smaller than 1, to allow  $R$  to change fast enough. In that case, the update (12) would be useless, and a stable method would be essential.

Are there equally efficient, stable methods? By (11) and (6)

$$\begin{aligned} z^H d &= x^H R^{-1} d \\ &= x^H w. \end{aligned}$$

Thus, (12) is equivalent to the formula

$$*w = \mu^{-1}w + \beta z x^H w. \tag{13}$$

Notice that  $w$  now appears twice. Perhaps the second use of  $w$  has stabilized the method? It has.

*Theorem:* The residual does not change when formula (13) is used: even when  $w$  only approximately satisfies  $Rw = d$ , the identity  $d - Rw = d - *R_*w$  holds.

*Proof:* It suffices to show that  $*R_*w = Rw$ . By direct computation

$$\begin{aligned} *R_*w &= (\mu R + (1 - \mu)xx^H)(\mu^{-1}w + \beta z x^H w) \\ &= Rw + xx^H w [\beta(\mu + (1 - \mu)x^H z) + (1 - \mu)\mu^{-1}] \\ &= Rw \end{aligned}$$

since  $\beta(\mu + (1 - \mu)x^H z) = (\mu - 1)\mu^{-1}$ .

This shows that the error is not amplified by (13). A similar analysis can be done for the method (12). It shows that the residual can increase by a scalar multiple of  $x$ , whose length is proportional to  $(x^H w - z^H d)$ .

An even more stable procedure can be devised. From (12) it follows that

$$*w - w \in \text{span} \{w, z\}. \tag{14}$$

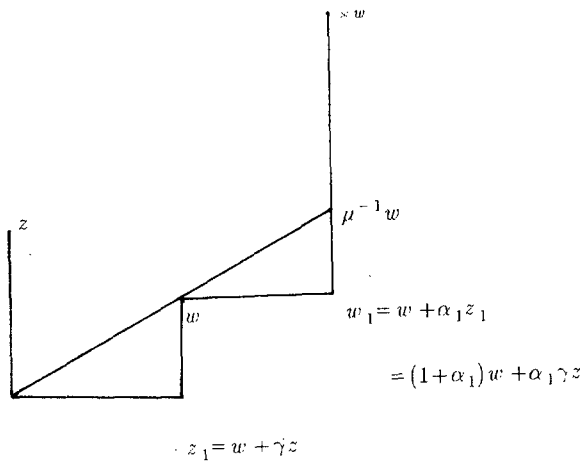


Fig. 1. Computation of  $\alpha_1$ .

All update procedures seek in some way to find the linear combination of  $w$  and  $z$  that, when added to  $w$ , gives  ${}^*w$ . Among the many possible methods are a group of conjugate direction procedures that are especially desirable in that they make almost no assumptions other than (14) and use the available data to choose the coefficients of  $w$  and  $z$  in the linear combination. In these methods, one chooses an orthogonal basis for the span of  $w$  and  $z$ , then takes a step from  $w$  in the direction of one of the basis vectors, going to the precise point in that direction closest to  ${}^*w$ . Then another step, in the direction of the other basis vector, produces  ${}^*w$ . The various possible algorithms differ in the choice of basis and the innerproduct used.

The following procedure is computationally convenient. Define  $z \equiv {}^*R^{-1}x$ . Note that  $z$  is a scalar multiple of  $R^{-1}x$ . Choose  $\gamma$  so that  $z$  and  $z_1 \equiv w - \gamma z$  are  ${}^*R$ -orthogonal. (Two vectors  $u$  and  $v$  are  ${}^*R$ -orthogonal if  $u^H {}^*Rv = 0$ .) Starting from  $w$ , take a step  $\alpha_1 z_1$  so that  $w_1 \equiv w + \alpha_1 z_1$  is as close as possible to  ${}^*w$  with respect to the  ${}^*R$ -norm. (The square of the  ${}^*R$ -norm of a vector  $v$  is given by  $v^H {}^*Rv$ .) Then take a step  $\alpha_2 z$  so that  $w_2 \equiv w_1 + \alpha_2 z$  is as close as possible to  ${}^*w$ . Now  ${}^*w = w_2$  is the updated solution.

It is well known [6] that if  $Rw = d$  exactly then, except for rounding errors,  $w_2$  exactly equals  ${}^*w$ . On the other hand, suppose that  $w = R^{-1}d + e$ , where  $e$  is the current error in  $w$ . We can write  $e = e_1 + e_2$ , where  $e_1 \in \text{span}\{w, z\}$  and  $e_2$  is  ${}^*R$ -orthogonal to both  $w$  and  $z$ . Then after application of the two conjugate direction steps as described above, we will obtain  ${}^*w = {}^*R^{-1}d + e_2$ . In other words, the component of the error that lies in the span of  $w$  and  $z$  will have been annihilated.

Computation of the step lengths in a conjugate direction method usually involves computing dot products. In this case, however, computation of  $\alpha_1$  can be greatly simplified by some geometric insight—see Fig. 1. By (12),  ${}^*w - \mu^{-1}w$  is a scalar multiple of  $z$ . Moreover,  $w_1 = (1 + \alpha_1)w + \alpha_1 \gamma z$ . Thus,  ${}^*w - w_1$  is a scalar multiple of  $z$  if  ${}^*w - (1 + \alpha_1)w$  is. Thus, we should take  $\alpha_1 = \mu^{-1} - 1$ .

Of course, the computed solution  $w$  does not exactly satisfy (12). So we should really compute  $\alpha_1$  by requiring

that  ${}^*w - w_1$  be  ${}^*R$ -orthogonal to  $z_1$ . But we would need to compute a matrix-vector product to be able to do this without error. That would raise the cost of the method from  $O(n)$  to  $O(n^2)$ . In fact, our procedure is equivalent to replacing the product  $Rw$  by the vector  $d$  in the inner-product  $z_1^H {}^*Rw$ . Let  $r = Rw - d$ . The error we make is therefore  $z_1^H r$ . Now  $z_1$  is orthogonal to  $x$ . So the error will be rather small if  $r$  is close to the span of  $x$ . In view of the fact that  $r$  is a residual and  $R$  is given by (1), it is likely that this is so.

The method is as follows.

*Algorithm (Conjugate Direction):* Given the Cholesky factor  $L$  of  $R$ , a new signal sample  $x$ , the current computed solution  $W = R^{-1}D$ , and  $\mu$ ,

- 1) update the Cholesky factor; now  ${}^*R = {}^*L {}^*L^H$ ;
- 2) solve  $({}^*L {}^*L^H)z = x$ ;
- 3) compute  $z^H x$  (which is real); and
- 4) for every column  $w$  of  $W$  and corresponding column  $d$  of  $D$ ,
  - a) compute  $x^H w$ ;
  - b) compute  $z^H d$ ;
  - c) compute  $\gamma := -x^H w / z^H x$ ;
  - d) compute  $z_1 := w + \gamma z$ ;
  - e) compute  $\alpha_1 := \mu^{-1} - 1$ ;
  - f) compute  $w_1 := w + \alpha_1 z_1$ ;
  - g) compute  $\alpha_2 := (z^H d - x^H w) / z^H x$ ;
  - h) compute  $w_2 := w_1 + \alpha_2 z$ ;
  - i) stop:  $w_2$  is the computed solution to  ${}^*R {}^*w = d$ .

Recall that computing  $x + \alpha y$  with complex  $x, y$ , and real  $\alpha$  costs one-half an operation. Thus, this conjugate direction procedure costs  $(\frac{5}{2})n^2 + (\frac{9}{2})nm$  operations. For  $m \gg n$  it is slightly less than twice as costly as the methods (12) and (13). In view of the experimental results given below, it does appear to be more accurate than (13) in some cases. But its superiority is not uniform; it depends on  $\sigma^2$  and  $\mu$ .

#### A. Experimental Tests

We have verified our claims by an experiment. Vectors  $x = (1, 2, 3, 4, 5, 6)^T + s$  were generated, where  $s$  had random, independent, normally distributed components of mean zero and variance  $\sigma^2$ . We took  $d = (1, 1, 1, 1, 1, 1)^T$  and initially  $R = I$  and  $w = d$ .

We then used the three methods discussed above for 100 updates. Let  $w^{(1)}$  denote the solution vector given by the method (12),  $w^{(2)}$  the solution given by the stable formula (13), and  $w^{(3)}$  the solution given by the conjugate direction formulas. At each step we updated the Cholesky factorization of  $R$  by computing

$$Q \begin{bmatrix} (1 - \mu)^{1/2} x^T \\ \mu^{1/2} L^T \end{bmatrix} = \begin{bmatrix} {}^*L^T \\ 0 \end{bmatrix}$$

where  $Q$  is the product of  $n$  plane rotations. We give four error statistics below. The first is a measure of the accuracy of the updated Cholesky factor  $L$ ,

$$E_R \equiv \|R - LL^T\|_F / \|R\|_F.$$

TABLE I  
RELATIVE ERRORS AFTER 100 UPDATES

$\mu = 0.8$	$\mu = 0.9$	$\mu = 0.99$
$\sigma^2 = 100$		
$E_R = 0.197(-6)$	$E_R = 0.501(-6)$	$E_R = 0.219(-5)$
$E_1 = 0.408(+3)$	$E_1 = 0.172(-1)$	$E_1 = 0.144(-4)$
$E_2 = 0.448(-5)$	$E_2 = 0.184(-5)$	$E_2 = 0.761(-5)$
$E_3 = 0.840(-6)$	$E_3 = 0.252(-6)$	$E_3 = 0.412(-6)$
$\sigma^2 = 10^{-2}$		
$E_R = 0.224(-6)$	$E_R = 0.307(-6)$	$E_R = 0.818(-6)$
$E_1 = 0.284(+4)$	$E_1 = 0.365(-1)$	$E_1 = 0.129(-4)$
$E_2 = 0.467(-5)$	$E_2 = 0.955(-5)$	$E_2 = 0.720(-5)$
$E_3 = 0.850(-5)$	$E_3 = 0.240(-5)$	$E_3 = 0.702(-6)$
$\sigma^2 = 10^{-6}$		
$E_R = 0.206(-6)$	$E_R = 0.677(-6)$	$E_R = 0.210(-5)$
$E_1 = 0.990(+5)$	$E_1 = 0.709(+0)$	$E_1 = 0.113(-4)$
$E_2 = 0.159(-2)$	$E_2 = 0.344(-3)$	$E_2 = 0.256(-4)$
$E_3 = 0.402(-2)$	$E_3 = 0.872(-4)$	$E_3 = 0.244(-4)$

A theoretical analysis of this procedure is given in Section II-B below. Also, for  $j = 1, 2, 3$ , we give a measure of the error in the updated solution  $w^{(j)}$ ,

$$E_j \equiv \|w^{(j)} - (LL^T)^{-1}d\| / \|(LL^T)^{-1}d\|.$$

We took  $\mu = 0.8, 0.9$ , and  $0.99$  and  $\sigma^2 = 10^2, 10^{-2}$ , and  $10^{-6}$ . The results were essentially unchanged for  $\sigma^2$  greater than  $10^2$ . In Table I we show the errors in the format

$$\begin{matrix} E_R \\ E_1 \\ E_2 \\ E_3 \end{matrix}$$

for each pair  $(\mu, \sigma^2)$ . The notation  $0.123(-4)$  means  $0.123 \times 10^{-4}$ . All computations were done in single precision on a VAX.

Note that the conjugate direction method (method 3) is distinctly more accurate in those cases where high accuracy is useful: low signal-to-noise ratio, which tends to make  $R$  well conditioned, and  $\mu \approx 1$ , so that  $R$  is accurately estimated. Three such cases occur in the upper-right-hand corner of Table I. In these cases, the conjugate direction method is ten times more accurate than the stable update method that uses (13).

**B. Another Stable Method**

We now discuss a third stable updating method that differs in two ways from those already considered. It avoids explicitly forming  $w$ ; and it can be viewed as an extension of the process for updating the Cholesky factor  $L$ —a process that we shall describe more fully in this section.

From (2) and (5) we have that

$$\begin{aligned} (\rho(d))^{-1} &= d^H R^{-1} d \\ &= v^H v \end{aligned} \tag{15}$$

where  $v$  is the solution to the triangular linear system

$$Lv = d. \tag{16}$$

And from (2), (4), and (3) we have that

$$\begin{aligned} g(d) &= w^H x \\ &= d^H R^{-1} x \rho(d) \\ &= (v^H y) \rho(d) \end{aligned} \tag{17}$$

where  $y$  is the solution to the triangular linear system

$$Ly = x. \tag{18}$$

If we are willing to solve the system (18) at a cost of  $n^2$  operations for every new signal  $x$  (and if there are many bearings  $d$ , this is reasonable), then we may use (17) to compute  $g(d)$ . Thus, we no longer need the weight vector  $w$ , but rather the vector  $v$  and the power estimate  $\rho$ . We now give a stable method for updating  $v$  and  $\rho$  after the change (7). This new algorithm is especially convenient in that it can be incorporated into the process of updating the Cholesky factor  $L$  of  $R$ . The systolic array devised by Schreiber and Tang [11] can be used to perform the necessary additional computations.

By (7), we seek the Cholesky factor  $*L$  of

$$\begin{aligned} *R &= [y, \mu^{1/2}L] \begin{bmatrix} y^H \\ \mu^{1/2}L^H \end{bmatrix} \\ &\equiv TT^H \end{aligned}$$

where  $y = (1 - \mu)^{1/2}x$ . Let  $Q$  be an  $n + 1 \times n + 1$  orthogonal matrix such that

$$TQ = [0 \quad *L] \tag{19}$$

where  $*L$  is lower triangular with positive real diagonal. It is easy to see that  $Q$  can be obtained as the product of  $n$  plane rotations. Now, clearly,

$$\begin{aligned} *R &= TQQ^H T^H \\ &= *L_*L^H \end{aligned}$$

so  $*L$  is the Cholesky factor of  $*R$ .

This method is very stable. If there is some error in  $L$ , for example, if

$$LL^H = R + E$$

where  $E$  is an error matrix, then by (7) and (19),

$$\begin{aligned} *L_*L^H &= TQQ^H T^H \\ &= TT^H \\ &= yy^H + \mu LL^H \\ &= *R + \mu E. \end{aligned}$$

Since  $\mu < 1$ , the error is reduced. In this sense, this method of updating the Cholesky factor is self-correcting. It was used in the experiments of the previous section, which show that it is very accurate. It can, therefore, be strongly recommended.

Now let  $d$  be a given steering vector, let  $\rho = \rho(d)$  be the corresponding power estimate, and let  $v = v(d) =$

$L^{-1}d$  be the corresponding solution to (16). Apply the rotations used in finding  $*L$  to obtain

$$[0, \mu^{-1/2}v^H]Q = [\bar{\delta}, *v^H]. \quad (20)$$

Now it follows, by (19) and (20), that

$$\begin{aligned} d &= [y \quad \mu^{1/2}L] \begin{bmatrix} 0 \\ \mu^{-1/2}v \end{bmatrix} \\ &= TQQ^H \begin{bmatrix} 0 \\ \mu^{-1/2}v \end{bmatrix} \\ &= [0 \quad *L] \begin{bmatrix} \delta \\ *v \end{bmatrix} \\ &= *L*v, \end{aligned}$$

so that  $*v$  is the updated solution to (16).

To show that (20) is both correct and stable, we assume that  $Lv$  is not exactly equal to  $d$ , but that

$$Lv = d + r$$

where  $r$  is a residual vector. But by (19) and (20),

$$\begin{aligned} d + r &= Lv \\ &= [y \quad \mu^{-1/2}L] \begin{bmatrix} 0 \\ \mu^{1/2}v \end{bmatrix} \\ &= [0 \quad *L] \begin{bmatrix} \delta \\ *v \end{bmatrix} \\ &= *L*v, \end{aligned}$$

so that  $*v$  satisfies (16) as well as did  $v$ . This is therefore a stable update method.

We now consider an efficient update formula for the power estimate  $\rho$ . From (20) it follows that

$$\begin{aligned} \mu^{-1}\rho^{-1} &= [0 \quad \mu^{-1/2}v^H] \begin{bmatrix} 0 \\ \mu^{-1/2}v \end{bmatrix} \\ &= [\bar{\delta} \quad *v^H] \begin{bmatrix} \delta \\ *v \end{bmatrix} \\ &= |\delta|^2 + *v^H*v \\ &= |\delta|^2 + *\rho^{-1}, \end{aligned}$$

so that the updated power estimate  $*\rho$  can be obtained.

Unfortunately, this last process is unstable. Suppose that

$$\rho^{-1} = v^Hv + \epsilon$$

where  $\epsilon$  is the error in  $\rho^{-1}$ . Then we have that

$$*\rho = [\mu^{-1}(v^Hv + \epsilon) - |\delta|^2]^{-1}.$$

But we know that  $|\delta|^2 + *v^H*v = \mu^{-1}v^Hv$  exactly, since (20) holds and  $Q^H Q = I$ . Thus,

$$*\rho = (*v^H*v + \mu^{-1}\epsilon)^{-1}$$

and

$$*\rho^{-1} = *v^H*v + \mu^{-1}\epsilon.$$

Since  $\mu^{-1} > 1$ , this approach is unstable, and  $*\rho$  should be computed as  $*v^H*v$ .

This procedure requires  $(\frac{3}{2})n^2$  operations for the Cholesky update,  $(\frac{1}{2})nm$  operations for updating  $*v$  using (20), and  $(\frac{1}{2})nm$  operations for recomputing  $*\rho$ .

### III. METHODS FOR UPDATING THE SPECTRAL DECOMPOSITION

A number of modern, high-resolution methods make use of the spectral decomposition of  $R$  [1], [9], [10]:

$$R = M\Lambda M^H \quad (21)$$

where  $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$  is the matrix of eigenvalues of  $R$ , ordered so that

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n.$$

Here  $M = [m_1, \dots, m_n]$  where  $m_i$  is a normalized eigenvector corresponding to  $\lambda_i$ . Note that  $M$  is unitary ( $M^H = M^{-1}$ ). We shall be concerned, therefore, with updating the decomposition (21) after a rank-one change (7) to  $R$ .

In practice, one uses an estimate of  $R$  that is the product  $X^H X$ , where  $X$  is a matrix whose rows are (weighted) samples of  $x$ . Therefore, the eigenvalues and eigenvectors of the estimate are the squared singular values and right singular vectors of  $X$ . Periodically, a new observation of  $x$  is made and is appended to  $X$  as a new row. Thus, the problem of updating the spectral decomposition (21) is mathematically equivalent to that of updating the singular values and right singular vectors of  $X$  when a row is appended.

Bunch and Nielsen recommend that to update the singular value decomposition of  $X$  when a row is added, one should update the corresponding spectral decomposition of  $R$  after a rank-one change [2]. This can obscure the small singular values of  $X$ : a singular value of size  $\sqrt{\epsilon}$  relative to the largest singular value becomes an eigenvalue of relative size  $\epsilon$ . If  $\epsilon$  is on the order of the machine precision, then this small eigenvalue may become zero. In some methods, these eigenvalues play an important role [9]; in others they do not [1]—only the eigenvectors or rather certain of the invariant subspaces do. When accurate determination of the small eigenvalues is necessary, the singular value approach is to be recommended.

We shall only consider updating the spectral decomposition (21). We derive two simplifications. First, we show that even when the signals are complex vectors, the bulk of the work in updating the decomposition consists in computing the spectral decomposition of a real symmetric matrix  $S$ . It has already been noted [7] that when the number of actual signal sources (including undesired sources), say,  $s$ , is less than the number of sensors in the array  $n$ , then we can work with an  $s + 1 \times s + 1$  matrix  $S_1$ . Moreover,  $S$  is the sum of a diagonal matrix and a matrix of rank one. Bunch, Nielsen, and Sorensen [3] give

a stable algorithm for finding the eigenvalues and eigenvectors of such a matrix in  $O(s^2)$  operations.

All the observations and methods proposed here have analogs for the SVD. In particular, Businger [4] has given a method that takes quadratic time for updating the SVD when a row is appended to the matrix. But the methods advocated in this section, including the eigenvalue updating algorithm, are all amenable to computation by systolic arrays, and the time for an update can be reduced in this way to  $O(n)$ . Part of Businger's method (the QR iteration for a bidiagonal matrix) is not.

Observe that, by (7) and (21),

$$*R = M[\mu\Lambda + (1 - \mu)zz^H]M^H,$$

where  $Mz = x$ . Let

$$S \equiv \mu\Lambda + (1 - \mu)zz^H$$

have the spectral decomposition

$$S = T_*\Lambda T^H, \tag{22}$$

and let

$$*M = MT; \tag{23}$$

then

$$*R = *M_*\Lambda_*M^H$$

is the desired spectral decomposition. Thus, the spectral decomposition (22) of the sum of a diagonal matrix and a matrix of rank one is needed.

We now show that  $S$  can be made real. Let  $w \in \mathbb{R}^n$  be given by  $w = (\omega_1, \dots, \omega_n)$ , where  $\omega_j = |\zeta_j|$ . Then

$$w = D^H z$$

where  $D = \text{diag}(\zeta_1/\omega_1, \dots, \zeta_n/\omega_n)$ . Note that  $DD^H = D^H D = I_n$ , i.e.,  $D$  is unitary. Now

$$\begin{aligned} w &= D^H z \\ &= D^H M^H x \\ &= (MD)^H x. \end{aligned}$$

The columns of  $MD$  are normalized eigenvectors of  $R$ . If we use them in place of the columns of  $M$ , we have that

$$S = \mu\Lambda + (1 - \mu)ww^T$$

which is real and symmetric. So we will assume in the following that  $M$  has been replaced by  $MD$ .

When  $R$  has repeated eigenvalues, there is more that can be done [3]. And if  $x$  is formed from  $s$  directional signals and spatially homogeneous, additive noise of power  $\sigma^2$ , then the eigenvalues of  $R$  satisfy

$$\lambda_s > \lambda_{s+1} = \lambda_{s+2} = \dots = \lambda_n = \sigma^2. \tag{24}$$

(The space  $N \equiv \text{span}\{m_{s+1}, \dots, m_n\}$  is called the noise subspace. While  $R$  determines  $N$ , any orthonormal basis for  $N$  can serve as the last  $n - s$  columns of  $M$ .) To take advantage of the repeated eigenvalue, let  $\Lambda_1 = \text{diag}(\lambda_1, \dots, \lambda_{s+1})$  and  $\Lambda_2 = \text{diag}(\lambda_{s+2}, \dots, \lambda_n)$ . It is possible to choose the last  $n - s$  columns of  $M$  so as to make  $\zeta_{s+2}$

$= \dots = \zeta_n = 0$ . If this is done, then

$$S = \begin{bmatrix} S_1 & 0 \\ 0 & \mu\Lambda_2 \end{bmatrix}$$

where  $S_1 = \mu\Lambda_1 + (1 - \mu)z_1 z_1^H$  and  $z_1 = (\zeta_1, \dots, \zeta_{s+1})^T$ . Now, given the spectral decomposition

$$S_1 = T_1 \Lambda_1 T_1^H,$$

we have that

$$S = \begin{bmatrix} T_1 & 0 \\ 0 & I_{n-s-1} \end{bmatrix} \begin{bmatrix} * \Lambda_1 & 0 \\ 0 & \mu \Lambda_2 \end{bmatrix} \begin{bmatrix} T_1^H & 0 \\ 0 & I_{n-s-1} \end{bmatrix}$$

is the spectral decomposition of  $S$ . This has been observed in previous work [7].

Let us be specific about the choice of  $m_{s+1}, \dots, m_n$ . Since  $z = M^H x$  we have

$$\zeta_i = m_i^H x, \quad 1 \leq i \leq s.$$

We must first compute these values. Now we let

$$m_{s+1} = \left( x - \sum_{i=1}^s \zeta_i m_i \right) / \left\| x - \sum_{i=1}^s \zeta_i m_i \right\|.$$

Thus,  $m_{s+1}$  is the normalized orthogonal projection of  $x$  on  $N$ . Now add additional vectors  $m_j, j = s + 2, \dots, n$ , until an orthonormal basis for  $N$  is obtained. This makes  $\zeta_j = 0$  for  $j = s + 2, \dots, n$ . In fact, it is not necessary to construct  $m_{s+2}, \dots, m_n$ . (If it were, they could be taken as the columns of a certain Householder matrix [3].)

Note that the number of eigenvalues (of the estimated covariance matrix) greater than  $\sigma^2$  can increase by 1 every time we update. On the other hand, the number of eigenvalues of  $R$  greater than  $\sigma^2$  is determined by the number of linearly independent signals hitting the array. The updating (7) moves one of the eigenvalues  $\sigma^2$  to the right, but the remainder move to the left, reduced by the factor  $\mu$ . In the equilibrium state of this process there are  $s$  eigenvalues greater than  $\sigma^2$  and a cluster of  $n - s$  near  $\sigma^2$ . This is not completely satisfactory.

Karasalo, Goetherstrom, and Westerlin suggest an attractive method that can be used if we have an *a priori* upperbound  $s$  on the number of signals [7]. They replace the new matrix  $*R$  by its closest approximation by a matrix of the form  $A + \sigma^2 I$ , where  $\text{rank}(A) = s$ . They show that

$$A = \sum_{i=1}^s *m_i * \lambda_i * m_i^H$$

and that the new noise level is

$$*\sigma^2 = [(n - s - 1)\sigma^2 + *\lambda_{s+1}]/(n - s).$$

To compute the eigenvectors of  $*R$ , note that

$$\begin{aligned} *M &= MT \\ &= [M_1, M_2] \begin{bmatrix} T_1 & 0 \\ 0 & I_{n-s-1} \end{bmatrix} \end{aligned} \tag{25}$$

where  $M_1 = [m_1, \dots, m_{s+1}]$ . Because  $T$  is real, this multiplication costs  $n(s+1)^2/2$  operations.

How much more costly is it to recompute (21) every step? Direct computation of a Hermitian spectral decomposition requires approximately  $5n^3 + O(n^2)$  operations. The work required by the method developed here is

1) form  $z_1$  at cost  $n(s+1)$  operations;  
 2) compute the spectral decomposition of  $S_1$  at cost  $O(s^2)$  operations; and

3) compute  $*M = M_1 T$  at cost  $(\frac{1}{2})(s+1)^2 n$  operations.

The net cost is therefore  $(\frac{1}{2})n(s+1)^2 + O(ns + s^2)$  operations. The relative expense of the new scheme drops very rapidly as  $n/s$  increases from 1, and is never more than 23 percent of the cost of a full spectral decomposition.

Additional computational savings are possible. Let us consider the methods proposed by Owsley [9], which are typical. These methods all seek to estimate the power of the signal hitting the array at a given angle by

$$g(d) = w^H d \quad (26)$$

where  $d$  is a steering vector for the given array and angle and

$$w = \sum_{j=1}^s m_j \beta(\lambda_j) m_j^H d \quad (27)$$

where  $\beta(\lambda)$  is real valued. Various choices for  $\beta$  can be made that succeed in suppressing the effect of noise and enhancing the resolution of the method.

In computing the output power  $g(d)$ , we can do better than to use the definition (26) directly. We can instead use the relations (25), (26), and (27) to reduce the cost from  $n^2$  operations to  $(s+1)^2/2$  operations for each vector  $d$ . To be specific, let

$$c = M^H d,$$

and

$$*c = *M^H d. \quad (28)$$

Denote the elements of these vectors by  $c = (\gamma_1, \dots, \gamma_n)^T$  and  $*c = (*\gamma_1, \dots, *\gamma_n)^T$ . Let  $c_1 \equiv (\gamma_1, \dots, \gamma_{s+1})^T$  and  $*c_1 \equiv (*\gamma_1, \dots, *\gamma_{s+1})^T$ . Then

$$g = \sum_{j=1}^{s+1} \beta(\lambda_j) |\gamma_j|^2$$

and

$$*g = \sum_{j=1}^{s+1} \beta(*\lambda_j) |*\gamma_j|^2. \quad (29)$$

To compute  $*g$ , we need only update  $c_1$ , then use (29). But by (25) and (28)

$$\begin{aligned} *c &= *M^H d \\ &= T^H M^H d \\ &= T^H c \\ &= \begin{bmatrix} T_1^H & 0 \\ 0 & I_{n-s-1} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \end{aligned}$$

TABLE II  
OPERATION COUNTS AND STABILITY OF THE BEAMFORMING ALGORITHMS

Method	Cost	Stability
Section II		
Nonrecursive solution of (4)	$n^2 m + nm + (3/2)n$	n.a.
Matrix inverse lemma: (10)-(11)-(12)	$(5/2)n[n+m]$	-
Stabilized matrix inversion: (10)-(11)-(13)	$(5/2)n[n+m]$	0
Conjugate direction:	$(1/2)n[5n+9m]$	+
Section II-B		
Updating the Cholesky factor $L$ using orthogonal transformations: (19)	$(3/2)n^2$	+
Updating the solution $v$ using the same orthogonal transformations (20):	$3nm$	+
Using the orthogonal transformations to update $\rho$ :	$m$	-
Recomputing $\rho = v^H v$ :	$nm$	n.a.
Key	n.a.	stability is not an issue for nonrecursive methods; the method is unstable: errors are amplified by every step;
	-	the method is stable: errors are neither amplified nor damped;
	0	the method is stable: errors are damped by every step;
	+	the method is stable: errors are damped by every step;

so that

$$*c_1 = T_1^H c_1. \quad (30)$$

Thus, from (29) and (30), we can compute  $g$  with only  $(s+1)^2/2$  operations for each vector  $d$ .

#### IV. CONCLUSIONS

We have given three numerically stable and computationally efficient procedures for adaptive beamforming that improve, either in speed or accuracy, on known procedures. These procedures make methods based on the inverse of the signal covariance matrix much more practical for real-time use. This is especially true for large sensor arrays, since the dominant cost of these procedures grows only linearly with the number of array elements (in this respect they are like the LMS method). Straightforward use of the matrix inverse or a triangular factorization incurs quadratic cost.

For methods based on a spectral decomposition of the signal covariance matrix, we have obtained a similar economy. The resulting rather dramatic reduction in cost makes these methods, too, more practical for real-time use.

To summarize the algorithms recommended, we give their operation counts and stability properties. In Table II, we give the results for the beamforming algorithms discussed in Section II. In Table III, we give the results for updating the spectral decomposition discussed in Section III.

We have not made an issue of stability of the spectral decomposition methods. Because no factor of  $\mu^{-1}$  occurs in the methods, there is no reason to suppose that insta-

TABLE III  
OPERATION COUNTS OF ALGORITHMS FOR THE SPECTRAL DECOMPOSITION

Method	Cost
Section III	
Full recomputation of the decomposition (21)	$5n^3 + 0(n^2)$
Use of the Bunch-Neilsen-Sorensen method (21)-(22)	$n^3 + 0(n^2)$
Exploiting repeated eigenvalues (24) when there are $s < n$ signals	$(1/2)s^2n + 0(ns + n^2)$
Recomputing $g(d)$ using the definition (26), when there are $m$ different direction vectors $d$	$mns$
Use of the recursive method (29)-(30) to compute $g(d)$	$(1/2)ms$

bility of the type encountered in the first beamforming method of Section II will occur. Moreover, Bunch, Neilsen, and Sorensen [3] and Karasalo, Goetherstrom, and Westerlin [7], [13] give substantial experimental evidence for the stability of some fast update methods for the spectral decomposition.

#### ACKNOWLEDGMENT

I thank P. Kuekes, N. Owsley, and G. Golub for many interesting discussions on this subject. I would also like to thank the referees, who made a number of very useful suggestions.

#### REFERENCES

- [1] G. Bienvenue and H. F. Mermoz, "New principle of array processing in underwater passive listening," in *VLSI and Modern Signal Processing*, S. Y. Kung, H. J. Whitehouse, and T. Kailath, Eds. Englewood Cliffs, NJ: Prentice-Hall, 1984.
- [2] J. R. Bunch and C. P. Nielsen, "Updating the singular value decomposition," *Numerische Mathematik*, vol. 31, pp. 111-130, 1978.
- [3] J. R. Bunch, C. P. Nielsen, and D. C. Sorensen, "Rank-one modification of the symmetric eigenproblem," *Numerische Mathematik*, vol. 31, pp. 31-48, 1978.

- [4] P. A. Businger, "Updating a singular value decomposition," *BIT*, vol. 10, pp. 376-385, 1970.
- [5] P. E. Gill, G. H. Golub, W. Murray, and M. A. Saunders, "Methods for modifying matrix factorizations," *Numerische Mathematik*, vol. 28, pp. 505-535, 1974.
- [6] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, MD: Johns Hopkins University Press, 1983.
- [7] I. Karasalo, L. Goetherstrom, and V. Westerlin, "Nagra signalbehandlingsformer basered pa kovariansestimater av signal och brus," (English title: "Some signal processing methods based on an estimate of the covariance of signal and noise") Swedish Defense Res. Inst. (FOA), Huvudavdelning 3, S581 11 Linkoeeping, Sweden, Tech. Rep. FOA Rep. C30343-E1, ISSN 0347-3708, Oct. 1983.
- [8] R. A. Monzingo and T. W. Miller, *Introduction to Adaptive Arrays*. New York: Wiley, 1980.
- [9] N. L. Owsley, "High resolution spectrum analysis by dominant mode enhancement," in *VLSI and Modern Signal Processing*, S. Y. Kung, H. J. Whitehouse, and T. Kailath, Eds. Englewood Cliffs, NJ: Prentice-Hall, 1984.
- [10] R. O. Schmidt, "A signal subspace approach to multiple emitter location and spectral estimation," Ph.D. dissertation, Stanford Univ., Stanford, CA, 1981.
- [11] R. Schreiber and W. P. Tang, "On systolic arrays for updating the Cholesky factorization," Swedish Roy. Inst. Technol., Dep. Numeric. Anal. Comput. Sci., Stockholm, Sweden, Tech. Rep. TRITANA-8313, 1983. also submitted to *BIT*.
- [12] G. W. Stewart, *Introduction to Matrix Computations*. New York: Academic, 1973.
- [13] I. Karasalo, "Estimating the covariance matrix by signal subspace averaging," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 8-12, Feb. 1986.



**Robert Schreiber** received the Ph.D. degree in computer science from Yale University, New Haven, CT, in 1977.

He has published research papers in numerical analysis, matrix computation, and special-purpose VLSI architectures. He has held teaching and research positions at the California Institute of Technology, Pasadena, and Stanford University, Stanford, CA, and was an architect of the Saxpy-IM scientific computer. He is currently an Associate Professor of Computer Science at Rensselaer

Polytechnic Institute, Troy, NY, and a Consultant to Saxpy Computer Corporation, Sunnyvale, CA.