An Adaptive Algorithm for Interference Cancelling in Array Processing

Franklin T. Luk

Department of Computer Science, Rensselaer Polytechnic Institute
Troy, New York 12180 USA

Sanzheng Qiao*

Department of Electronic Engineering, City University of Hong Kong
Tat Chee Avenue, Kowloon, Hong Kong

ABSTRACT

In array processing, one technique for cancelling interference in the presence of colored noise is the $ULLV$ decomposition of a pair of matrices. The factorization is stable and accurate, and is easy to update when a row is added to either one of the two matrices. In earlier work, we made the assumptions that both matrices must have at least as many rows as columns and that the matrix representing interference must have full column rank. In this paper, we relax the latter restriction and allow the interference matrix to be rank deficient. We present an algorithm for computing the decomposition and two techniques for updating it. The details on updating are quite different from those in the full rank case.

Keywords: $ULLV$ decomposition, matrix updating, noise subspace tracking.

1. INTRODUCTION

Estimating the noise subspace is an important problem in signal processing. When the noise is colored and data updating is necessary, a special matrix decomposition is required. In [2] we introduced a new decomposition of a pair of matrices that we named the $ULLV$ decomposition. Call the matrices A and B, and let A be $n \times p$ and B be $m \times p$ (note that p represents the number of parameters). Due to the application, it is always true that $n \geq p$. In [2] we assumed that $m \geq p$ and B has rank p. In this paper, we complete our work by addressing the $ULLV$ decomposition when B has column rank less than p. We will present an algorithm and two updating schemes. Since $\mathrm{Null}(B) \neq \{0\}$, the updating schemes, especially the procedure for updating B, are substantially different from those in [2].

This paper is organized as follows. We describe the application of the $ULLV$ decomposition in interference cancelling in Section 2. In Section 3, we discuss the $ULLV$ decomposition of A and B when the latter is rank deficient, and we show how to compute the noise subspace. We describe a technique to propagate rotations in Section 4. Updating schemes are presented in Sections 5 and 6.

* Permanent Address: Department of Computer Science and Systems, McMaster University, Hamilton, Ontario L8S 4K1 CANADA

2. ESTIMATING NOISE SUBSPACE

Noise subspace estimation is an important problem in signal processing. For example, in the MUSIC (MUltiple SIgnal Classification) setting, the received signal-plus-noise vector $x \in C^p$ is formulated as

$$x = Fs + v, \tag{2.1}$$

where F $(p \times k,\ p \geq k)$ contains information about the arrival angles (or frequencies), s denotes the signal vector, and v designates the noise vector. Let $A^H A$ $(A \in C^{n \times p},\ n \geq p)$ be an approximation of the correlation matrix $E[xx^H]$. Then A represents a signal-plus-noise matrix.

When the noise is white, the noise correlation matrix $E[vv^H]$ is a scalar multiple of the identity matrix. The noise subspace can be computed using the singular value decomposition (SVD). Specifically, let

$$A = UDV^H$$

denote an SVD of A, where U $(n \times p)$ is orthonormal, V $(p \times p)$ is unitary, and $D = \mathrm{diag}(d_i)$ is diagonal, with

$$d_1 \geq d_2 \geq \cdots \geq d_p \geq 0.$$

To find the noise subspace, we determine k, the rank of F in (2.1), so that

$$d_k \gg d_{k+1} \geq \cdots \geq d_p \geq 0. \tag{2.2}$$

The last $p - k$ columns of V form an orthonormal basis for the noise subspace [3], [4]. The SVD approach is numerically sound. Unfortunately, it is not amendable to updating when a row is added to A. To overcome this obstacle, Stewart [5] proposed an updatable rank-revealing factorization that he called the ULV decomposition:

$$A = ULV^H,$$

where U is orthonormal and V is unitary. The matrix L is lower triangular with the rank-revealing structure

$$L = \begin{pmatrix} \tilde{L} & 0 \\ E & G \end{pmatrix},$$

where $\tilde{L}$ is a "large" block in that

$$\sigma_{\min}(\tilde{L}) \approx d_k,$$

and E and G are "small" blocks in that

$$\|E\|_F^2 + \|G\|_F^2 \approx d_{k+1}^2 + \cdots + d_p^2.$$

The quantity $\sigma_{\min}(\tilde{L})$ denotes the smallest singular value of $\tilde{L}$. The subspace spanned by the last $p - k$ columns of V thus approximates the subspace associated with the $p - k$ smallest singular values of A. In this application, our primary interest concerns the noise subspace, and not the individual singular values nor singular vectors. The ULV decomposition effectively approximates the subspace corresponding to the smallest singular values, namely, the noise subspace. A major advantage of the ULV decomposition over the SVD is that it can be efficiently updated when a row is added to A.

Now, suppose that the noise is colored. Let $B^H B$ approximate the correlation matrix of the colored noise. Assume that B $(m \times p,\ m \geq p)$ has full column rank. Then we use the generalized singular value decomposition (GSVD) [6] of A and B:

$$A = U_A D_A L V^H \quad \text{and} \quad B = U_B L V^H,$$

where U_A $(n \times p)$ and U_B $(m \times p)$ are orthonormal, V $(p \times p)$ is unitary, L $(p \times p)$ is lower triangular and nonsingular, and D_A is diagonal $(p \times p)$. The diagonal elements of D_A, called the generalized singular values, are arranged in a nonincreasing order. Suppose that these values satisfy the relation (2.2). Then the last $p - k$ columns of V form an orthonormal basis for the noise subspace.

3. THE $ULLV$ DECOMPOSITION

In a generalized SVD of A and B, the underlying problem is the so called symmetric pencil [1], defined by $A^H A - \sigma^2 B^H B$. Specifically, if $\sigma \geq 0$ and $x \neq 0$ satisfy

$$(A^H A - \sigma^2 B^H B)x = 0, \tag{3.1}$$

then σ denotes a generalized singular value of (A, B) and x represents the corresponding generalized singular vector. We assume that

$$\mathrm{Null}(A) \cap \mathrm{Null}(B) = \{0\}. \tag{3.2}$$

Otherwise, there exists some $y \neq 0$ so that

$$Ay = By = 0,$$

and there will be infinitely many σ satisfying (3.1). Previously, we assumed that B is $m \times p$, $m \geq p$, and $B^H B$ is nonsingular. In this paper, we consider the case where $B^H B$ is rank deficient. One may argue that in real life, the noise correlation matrix $B^H B$ is always nonsingular due to the presence of noise. Then why do we consider this singular case? We answer as follows: in addition to the obvious academic interest imposed by a singular $B^H B$, the presence of *very strong jammers* may provide a good reason to consider the matrix as *numerically singular.*

Let us show now how we may assume that B is $m \times p$ with full row rank. Suppose that B is $l \times p$ and that $B^H B$ $(p \times p)$ is rank deficient, say with

$$\mathrm{rank}(B^H B) = m < p.$$

Let

$$B = U \Sigma V^H$$

be an SVD of B. We partition:

$$\Sigma = \begin{pmatrix} \hat{\Sigma} & 0 \\ 0 & 0 \end{pmatrix},$$

where $\hat{\Sigma}$ is $m \times m$ and nonsingular. Thus,

$$B^H B = V \begin{pmatrix} \hat{\Sigma}^2 & 0 \\ 0 & 0 \end{pmatrix} V^H = V \begin{pmatrix} \hat{\Sigma} \\ 0 \end{pmatrix} \begin{pmatrix} \hat{\Sigma} & 0 \end{pmatrix} V^H.$$

Define a new $m \times p$ matrix $\hat{B}$ by

$$\hat{B} = \begin{pmatrix} \hat{\Sigma} & 0 \end{pmatrix} V^H.$$

Then $\hat{B}$ has full row rank, and $\hat{B}^H \hat{B} = B^H B$. Thus, without loss of generality, we may replace B by $\hat{B}$.

From here on, we shall assume that B is $m \times p$ with $m < p$, and that B has full row rank. We compute the LQ decomposition of B:

$$B = (L \quad 0) V^H,$$

where V $(p \times p)$ is unitary and L $(m \times m)$ is lower triangular and nonsingular. Suppose that A is $n \times p$ $(n \geq p)$. We perform a QL decomposition of the product AV:

$$AV = U_A L_A,$$

where U_A $(n \times p)$ is orthonormal and L_A $(p \times p)$ is lower triangular. Partition the triangular matrix product:

$$L_A \begin{pmatrix} L^{-1} & 0 \\ 0 & I_{p-m} \end{pmatrix} = \begin{pmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{pmatrix},$$

where L_{11} is $m \times m$. We get the following decompositions:

$$A = U_A \begin{pmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{pmatrix} \begin{pmatrix} L & 0 \\ 0 & I_{p-m} \end{pmatrix} V^H \tag{3.3a}$$

and

$$B = (I_m \ \ 0) \begin{pmatrix} L & 0 \\ 0 & I_{p-m} \end{pmatrix} V^H = (L \ \ 0) V^H.$$

The latter can be generalized to

$$B = U_B (I_m \ \ 0) \begin{pmatrix} L & 0 \\ 0 & I_{p-m} \end{pmatrix} V^H = U_B (L \ \ 0) V^H, \tag{3.3b}$$

where U_B ($m \times m$) is unitary. For now, U_B is initialized as an identity matrix. We claim that L_{22} is nonsingular. Indeed, if $L_{22}x = 0$ for some $x \neq 0$, then it can be readily verified that the nontrivial vector y, defined by

$$y = V \begin{pmatrix} 0 \\ x \end{pmatrix},$$

will satisfy $Ay = By = 0$, contradicting the assumption (3.2). Thus, defining a $p \times m$ matrix X by

$$X = V \begin{pmatrix} L^{-1} \\ -L_{22}^{-1} L_{21} \end{pmatrix},$$

we get from (3.3a) that

$$A^H A X = V \begin{pmatrix} L & 0 \\ 0 & I_{p-m} \end{pmatrix}^H \begin{pmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{pmatrix}^H \begin{pmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{pmatrix} \begin{pmatrix} I_m \\ -L_{22}^{-1} L_{21} \end{pmatrix} = V \begin{pmatrix} L^H \\ 0 \end{pmatrix} L_{11}^H L_{11}, \tag{3.4}$$

and from (3.3b) that

$$B^H B X = V \begin{pmatrix} L^H \\ 0 \end{pmatrix} (L \ \ 0) \begin{pmatrix} L^{-1} \\ -L_{22}^{-1} L_{21} \end{pmatrix} = V \begin{pmatrix} L^H \\ 0 \end{pmatrix}. \tag{3.5}$$

Consider an SVD of L_{11}:

$$L_{11} = \tilde{U} \Sigma \tilde{V}^H, \tag{3.6}$$

and define Z by $Z = X\tilde{V}$. From (3.4) and (3.5), we get

$$A^H A Z = B^H B Z \Sigma^2.$$

Following the definition of σ in (3.1), the above equation shows that the m singular values of L_{11} are the generalized singular values of (A, B) and that the m columns of Z are the corresponding generalized singular vectors. As noted before, we are interested in finding the noise subspace. Thus, the SVD of (3.6) can be replaced by a rank-revealing ULV decomposition. If (A, B) has k "large" generalized singular values, then the last $m - k$ columns of Z form the noise subspace.

A crucial step in the ULV decomposition is deflation. The workhorses of the deflation are the left and right rotations [5]. When we perform a ULV decomposition on the block L_{11}, we must keep the structure of the decompositions (3.3a) and (3.3b) using rotations. This leads to the following definition of the $ULLV$ decomposition ($ULLVD$) and an algorithm for its computation.

Definition ULLVD ($m < p$).

The $ULLV$ decomposition of a pair of matrices A ($n \times p$, $n \geq p$) and B ($m \times p$, $m < p$, full row rank) is

$$A = U_A \begin{pmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{pmatrix} \begin{pmatrix} L & 0 \\ 0 & I_{p-m} \end{pmatrix} V^H \tag{3.7a}$$

and

$$B = U_B (I_m \;\; 0) \begin{pmatrix} L & 0 \\ 0 & I_{p-m} \end{pmatrix} V^H, \tag{3.7b}$$

where U_A ($n \times p$) is orthonormal, and U_B ($m \times m$) and V ($p \times p$) are unitary. The blocks L ($m \times m$), L_{11} ($m \times m$), and L_{22} ($(p-m) \times (p-m)$) are lower triangular, and L_{11} has the rank-revealing structure:

$$L_{11} = \begin{pmatrix} \tilde{L} & 0 \\ E & G \end{pmatrix}, \tag{3.7c}$$

where

(i) $\tilde{L}$ is the "large" block in that $\sigma_{\min}(\tilde{L}) \approx d_k$;

(ii) G and E are "small" blocks in that $||E||^2 + ||G||^2 \approx d_{k+1}^2 + \cdots + d_m^2$. □

In the above definition, d_i denotes the ith singular value of L_{11}, and thus the ith generalized singular value of (A, B). If the detected rank of L_{11} is k, then the last $m-k$ columns of the matrix

$$V \begin{pmatrix} L^{-1} \\ -L_{22}^{-1} L_{21} \end{pmatrix} \tag{3.8}$$

form the noise subspace. When $m \geq p$, the blocks L_{21} and L_{22} in the above definition vanish, and both L_{11} and L are $p \times p$. Consequently, the last $p-k$ columns of V form an orthonormal basis for the noise subspace, since L^{-1} is lower triangular and the lower block $-L_{22}^{-1} L_{21}$ vanishes.

Algorithm ULLVD ($m < p$).

1 . Compute an LQ decomposition of B: $B = (L \;\; 0) V^H$,

   or equivalently, compute a QR decomposition of B^H: $B^H = V(L \;\; 0)^H$;

2 . Compute a QL decomposition of AV: $AV = U_A L_A$;

3 . Partition the matrix product:

$$L_A \begin{pmatrix} L^{-1} & 0 \\ 0 & I \end{pmatrix} = \begin{pmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{pmatrix};$$

4 . Perform the ULV decomposition of L_{11} while propagating the rotations carefully (see §4);

5 . Calculate the decompositions (3.7a) and (3.7b), and obtain the noise subspace via (3.8). □

4. PROPAGATING ROTATIONS

The building blocks of our algorithms are plane rotations. Since we deal with the products of structured matrices, it is necessary to maintain the products as well as the structures of the matrix factors when a rotation is applied to one of the factors in the product. Consider the product AB of lower triangular matrices A and B. When a right rotation P is applied to the columns of A, it becomes necessary to apply P^H to the corresponding rows of B in order to keep the product unchanged; that is,

$$AB = (AP)(P^H B).$$

Figure 4.1 illustrates the procedure. The entries of A are represented by 'a' and those of B by 'b'. Those columns of A to which the rotation P is applied are indicated by a pair of down arrows.

$$\begin{array}{c} \begin{array}{cccc} & \downarrow & \downarrow & \end{array} \\ \begin{pmatrix} a & & & \\ a & a & & \\ a & a & a & \\ a & a & a & a \end{pmatrix} \end{array} P_{2,3}P_{2,3}^H \begin{pmatrix} b & & & \\ b & b & & \\ b & b & b & \\ b & b & b & b \end{pmatrix} = \begin{pmatrix} a & & & \\ a & a & a & \\ a & a & a & \\ a & a & a & a \end{pmatrix} \begin{pmatrix} b & & & \\ b & b & b & \\ b & b & b & \\ b & b & b & b \end{pmatrix}$$

Figure 4.1. Effect of rotation $P_{2,3}$.

As shown in Figure 4.1, after the application of the rotation, although the product is unchanged, the triangular structure of A and B is altered. To restore the structure, we apply a left rotation P_A^H to the rows of A and a right rotation P_B to the columns of B (illustrated in Figure 4.2). Those rows of A are indicated by a pair of right arrows and those columns of B are indicated by a pair of down arrows. Thus, we get

$$A = P_A(P_A^H A) \quad \text{and} \quad B = (BP_B)P_B^H.$$

Combining the above two procedures, we obtain

$$AB = (AP)(P^H B) = P_A \hat{A}\hat{B} P_B^H,$$

where the matrices

$$\hat{A} = P_A^H AP \quad \text{and} \quad \hat{B} = P^H BP_B$$

are lower triangular. This is how we can maintain the product and the structure by propagating a rotation P to the left through A and to the right through B. Clearly, this technique can be generalized to a product of more than two matrices. A rotation applied to a matrix in the product can be propagated until it reaches the end. All rotations at one end can be absorbed into one unitary matrix. So,

$$UA = \tilde{U}(P_A^H A) \quad \text{or} \quad BV^H = (BP_B)\tilde{V}^H,$$

where $\tilde{U} = UP_A$ and $\tilde{V} = VP_B$.

$$\begin{array}{c} \\ \\ \rightarrow \\ \rightarrow \\ \\ \end{array} \begin{pmatrix} a & & & \\ a & a & a & \\ a & a & a & \\ a & a & a & a \end{pmatrix} \begin{array}{c} \begin{array}{cccc} & \downarrow & \downarrow & \end{array} \\ \begin{pmatrix} b & & & \\ b & b & b & \\ b & b & b & \\ b & b & b & b \end{pmatrix} \end{array} = P_A \begin{pmatrix} a & & & \\ a & a & & \\ a & a & a & \\ a & a & a & a \end{pmatrix} \begin{pmatrix} b & & & \\ b & b & & \\ b & b & b & \\ b & b & b & b \end{pmatrix} P_B^H$$

Figure 4.2. Restoration of the triangular structure.

5. ADDING A ROW TO A

In this section, we present a scheme for updating the $ULLV$ decomposition when a row is added to A. We emphasize that special care must be taken to maintain the structures of the decompositions of (3.7a), (3.7b), and (3.7c). Specifically, L_{11}, L_{22}, and L should stay lower triangular, I_m I_{p-m} remain as identity matrices, and L_{11} must retain the rank-revealing structure (3.7c).

Suppose that we have the $ULLV$ decomposition, and we add a row vector a^T to A. Denoting

$$(u^T \quad v^T) = a^T V \begin{pmatrix} L^{-1} & 0 \\ 0 & I_{p-m} \end{pmatrix},$$

where u is $m \times 1$ and v is $(p-m) \times 1$, we have

$$\begin{pmatrix} A \\ a^T \end{pmatrix} = \begin{pmatrix} U_A & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} L_{11} & 0 \\ L_{21} & L_{22} \\ u^T & v^T \end{pmatrix} \begin{pmatrix} L & 0 \\ 0 & I_{p-m} \end{pmatrix} V^H. \tag{5.1a}$$

Clearly, a forgetting factor λ $(0 < \lambda \leq 1)$ can be readily integrated into the above decomposition. In particular,

$$\begin{pmatrix} \lambda A \\ a^T \end{pmatrix} = \begin{pmatrix} U_A & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \lambda L_{11} & 0 \\ \lambda L_{21} & \lambda L_{22} \\ u^T & v^T \end{pmatrix} \begin{pmatrix} L & 0 \\ 0 & I_{p-m} \end{pmatrix} V^H.$$

For simplicity, in the following discussion, we assume $\lambda = 1$. The decomposition (3.7b) of B is unchanged:

$$B = U_B (I_m \quad 0) \begin{pmatrix} L & 0 \\ 0 & I_{p-m} \end{pmatrix} V^H. \tag{5.1b}$$

For convenience, we refer to the first and the second matrix factors in the decomposition (5.1a) as $\tilde{U}_A$ and $\tilde{L}_A$, respectively:

$$\tilde{U}_A = \begin{pmatrix} U_A & 0 \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad \tilde{L}_A = \begin{pmatrix} L_{11} & 0 \\ L_{21} & L_{22} \\ u^T & v^T \end{pmatrix}.$$

We eliminate v^T in the above $\tilde{L}_A$ using left rotations on the last $p - m + 1$ rows of $\tilde{L}_A$. These rotations will be absorbed into $\tilde{U}_A$. We introduce zeros to all but the first element of u^T in $\tilde{L}_A$ using right rotations on the first m columns of $\tilde{L}_A$. Each right rotation is followed by a left rotation to restore the triangular structure as illustrated in Figure 5.1 with $m = 4$. The figure shows how the lower triangular structure of L_{11} is maintained. Also, the size of the "large" block, whose entries are indicated by 'L', in L_{11} is increased by only one. As described earlier, in the decomposition (5.1a), a right rotation on L_{11} is propagated to the right and a left rotation on L_{11} is absorbed by $\tilde{U}_A$. In (5.1b), a left rotation on L can be absorbed by U_B. A right rotation on L can be absorbed by V. Finally, since the rotations are applied to the first m columns and the first m rows of $\tilde{L}_A$, the identity matrices in the decompositions (5.1a) and (5.1b) are not affected. At the end of this step, the decomposition (5.1a) has the following structure:

$$\begin{pmatrix} A \\ a^T \end{pmatrix} = \tilde{U}_A \begin{pmatrix} L & 0 & 0 & 0 & 0 \\ L & L & 0 & 0 & 0 \\ L & L & L & 0 & 0 \\ s & s & s & s & 0 \\ z & z & z & z & \times \\ x & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \times & 0 & 0 & 0 & 0 \\ \times & \times & 0 & 0 & 0 \\ \times & \times & \times & 0 & 0 \\ \times & \times & \times & \times & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tilde{V}^H.$$

We annihilate the entire last row of the second matrix in the above decomposition using a left rotation on its first and last rows. After updating, the rank may change (increase/decrease) and deflation may become necessary. Since the size of the "large" block in L_{11} increases by only one, at most two deflation steps will be required.

$$
\begin{array}{cccc}
 & & \downarrow & \downarrow \\
L & & & \\
L & L & & \\
s & s & s & \\
s & s & s & s \\
z & z & z & z \\
x & x & x & x
\end{array}
\Longrightarrow
\begin{array}{c}
 \\ \\ \\ \rightarrow \\ \rightarrow \\ \\ \\
\end{array}
\begin{array}{cccc}
 & & & \\
L & & & \\
L & L & & \\
s & s & s & s \\
s & s & s & s \\
z & z & z & z \\
x & x & x &
\end{array}
\Longrightarrow
\begin{array}{cccc}
 & \downarrow & \downarrow & \\
L & & & \\
L & L & & \\
s & s & s & \\
s & s & s & s \\
z & z & z & z \\
x & x & x &
\end{array}
\Longrightarrow
\begin{array}{c}
 \\ \\ \rightarrow \\ \rightarrow \\ \\ \\ \\
\end{array}
\begin{array}{cccc}
 & & & \\
L & & & \\
L & L & L & \\
s & s & s & \\
s & s & s & s \\
z & z & z & z \\
x & x & &
\end{array}
$$

$$
\Longrightarrow
\begin{array}{cccc}
\downarrow & \downarrow & & \\
L & & & \\
L & L & & \\
L & L & L & \\
s & s & s & s \\
z & z & z & z \\
x & x & &
\end{array}
\Longrightarrow
\begin{array}{c}
 \\ \rightarrow \\ \rightarrow \\ \\ \\ \\ \\
\end{array}
\begin{array}{cccc}
 & & & \\
L & L & & \\
L & L & & \\
L & L & L & \\
s & s & s & s \\
z & z & z & z \\
x & & &
\end{array}
\Longrightarrow
\begin{array}{cccc}
 & & & \\
L & & & \\
L & L & & \\
L & L & L & \\
s & s & s & s \\
z & z & z & z \\
x & & &
\end{array}
$$

Figure 5.1. Procedure to eliminate all but the first element of the last row of $\tilde{L}_A$, where 'L' and 's' denote "large" and small entries in L_{11}, resp., 'z' designates an entry in L_{21}, and 'x' an element of u.

Algorithm Updating ULLVD (add a row to A).

1 . Eliminate v^T in the above $\tilde{L}_A$ using left rotations on the last $p-m+1$ rows of $\tilde{L}_A$.

2 . Introduce zeros to all but the first element of u^T in $\tilde{L}_A$ using right rotations on the first m columns of $\tilde{L}_A$.

3 . Annihilate the entire last row of the second matrix using a left rotation on its first and last rows.

4 . After updating, the rank may change. Use deflation if necessary to update the rank. □

6. ADDING A ROW TO B

Now, we add a row b^T to B. Partition $b^T V = (x^T \quad y^T)$, where x is $m \times 1$ and y is $(p-m) \times 1$. We write

$$A = U_A \begin{pmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \end{pmatrix} \begin{pmatrix} L & 0 \\ 0 & I_{p-m} \\ x^T & y^T \end{pmatrix} V^H \tag{6.1a}$$

and

$$\begin{pmatrix} B \\ b^T \end{pmatrix} = \begin{pmatrix} U_B & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} I_m & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} L & 0 \\ 0 & I_{p-m} \\ x^T & y^T \end{pmatrix} V^H. \tag{6.1b}$$

For simplicity, we shall assume that $\lambda = 1$. But we can easily accommodate a forgetting factor λ as follows:

$$A = U_A \begin{pmatrix} \lambda^{-1} L_{11} & 0 & 0 \\ \lambda^{-1} L_{21} & L_{22} & 0 \end{pmatrix} \begin{pmatrix} \lambda L & 0 \\ 0 & I_{p-m} \\ x^T & y^T \end{pmatrix} V^H$$

and

$$\begin{pmatrix} \lambda B \\ b^T \end{pmatrix} = \begin{pmatrix} U_B & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} I_m & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \lambda L & 0 \\ 0 & I_{p-m} \\ x^T & y^T \end{pmatrix} V^H.$$

Our goal is to eliminate x and y and update both (6.1a) and (6.1b). In the following steps, we show how to achieve this goal through a careful elimination process. To illustrate the procedure, we show the structure of the decompositions (6.1a) and (6.1b) for $m = 3$ and $p = 6$.

$$A = U_A \begin{pmatrix} \times & 0 & 0 & 0 & 0 & 0 & 0 \\ \times & \times & 0 & 0 & 0 & 0 & 0 \\ \times & \times & \times & 0 & 0 & 0 & 0 \\ \times & \times & \times & \times & 0 & 0 & 0 \\ \times & \times & \times & \times & \times & 0 & 0 \\ \times & \times & \times & \times & \times & \times & 0 \end{pmatrix} \begin{pmatrix} \times & 0 & 0 & 0 & 0 & 0 \\ \times & \times & 0 & 0 & 0 & 0 \\ \times & \times & \times & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \times & \times & \times & \times & \times & \times \end{pmatrix} V^H \tag{6.2a}$$

and

$$\begin{pmatrix} B \\ b^T \end{pmatrix} = \begin{pmatrix} U_B & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} I_m & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \times & 0 & 0 & 0 & 0 & 0 \\ \times & \times & 0 & 0 & 0 & 0 \\ \times & \times & \times & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \times & \times & \times & \times & \times & \times \end{pmatrix} V^H. \tag{6.2b}$$

First, we use right rotations to transform y^T in (6.1a) and (6.1b) into $(\eta, 0, \ldots, 0)$. To maintain the identity part in the third matrix factor, these rotations are propagated to the left in the both decompositions (6.1a) and (6.1b). At the end of this step, the third matrix factor has the following structure:

$$\begin{pmatrix} \times & 0 & 0 & 0 & 0 & 0 \\ \times & \times & 0 & 0 & 0 & 0 \\ \times & \times & \times & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \times & \times & \times & \times & 0 & 0 \end{pmatrix}.$$

Our second step is to permute the $(m+1)$-st and the last rows of the above matrix in the two decompositions (6.2a) and (6.2b). We propagate this permutation to the left and get

$$A = U_A \begin{pmatrix} \times & 0 & 0 & 0 & 0 & 0 & 0 \\ \times & \times & 0 & 0 & 0 & 0 & 0 \\ \times & \times & \times & 0 & 0 & 0 & 0 \\ \times & \times & \times & 0 & 0 & 0 & \times \\ \times & \times & \times & 0 & \times & 0 & \times \\ \times & \times & \times & 0 & \times & \times & \times \end{pmatrix} \begin{pmatrix} \times & 0 & 0 & 0 & 0 & 0 \\ \times & \times & 0 & 0 & 0 & 0 \\ \times & \times & \times & 0 & 0 & 0 \\ \times & \times & \times & \times & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} V^H \tag{6.3a}$$

and

$$\begin{pmatrix} B \\ b^T \end{pmatrix} = \begin{pmatrix} U_B & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} I_{m+1} & 0 \end{pmatrix} \begin{pmatrix} \times & 0 & 0 & 0 & 0 & 0 \\ \times & \times & 0 & 0 & 0 & 0 \\ \times & \times & \times & 0 & 0 & 0 \\ \times & \times & \times & \times & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} V^H. \tag{6.3b}$$

The third step is to transform the last row of the third matrix in (6.3a) and (6.3b) into $(1, 0, \ldots, 0)$ using a sequence of column permutations. As before, these permutation are absorbed by V. To restore the triangular structure, each permutation is also propagated to the left. Figure 6.1 shows one such step. In (6.3b), a left rotation on the third matrix factor can be propagated to the left and absorbed into U_B due to the presence of I_{m+1} in the second matrix. In (6.3a), a left rotation on the third matrix factor must be propagated to the left through the second matrix and then absorbed into U_A. Figure 6.2 shows the propagation of one such rotation and the change of the structure of the second matrix in (6.3a). We get

$$A = U_A \begin{pmatrix} \times & 0 & 0 & 0 & 0 & 0 & \times \\ \times & \times & 0 & 0 & 0 & 0 & \times \\ \times & \times & \times & 0 & 0 & 0 & \times \\ \times & \times & \times & \times & 0 & 0 & \times \\ \times & \times & \times & \times & \times & 0 & \times \\ \times & \times & \times & \times & \times & \times & \times \end{pmatrix} \begin{pmatrix} \times & 0 & 0 & 0 & 0 & 0 \\ \times & \times & 0 & 0 & 0 & 0 \\ \times & \times & \times & 0 & 0 & 0 \\ \times & \times & \times & \times & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} V^H \tag{6.4a}$$

and

$$\begin{pmatrix} B \\ b^T \end{pmatrix} = \begin{pmatrix} U_B & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} I_{m+1} & 0 \end{pmatrix} \begin{pmatrix} \times & 0 & 0 & 0 & 0 & 0 \\ \times & \times & 0 & 0 & 0 & 0 \\ \times & \times & \times & 0 & 0 & 0 \\ \times & \times & \times & \times & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} V^H. \tag{6.4b}$$

Our fourth step is to use a left rotation on the first and last rows of the third matrix in (6.4a) and (6.4b) to zero out the entire last row. In (6.4a), this rotation is propagated to the left. Now that the last row of the third matrix is zero, we can drop the last column of the second matrix. In (6.4b), the rotation propagation changes the second matrix into

$$\begin{pmatrix} \times & 0 & 0 & \times \\ 0 & I_m & 0 & 0 \end{pmatrix}.$$

As in (6.4a), the last column of the above matrix can be dropped. Then the resulting matrix can be restored to $(I_{m+1} \;\; 0)$ by scaling its first column. In order to keep the products unchanged, we scale the corresponding first row of the third matrix in (6.4a) and (6.4b) and the first column of the second matrix in (6.4a). This completes the restoration of the $ULLV$ decomposition structure after a row is added to B. Finally, when a row is added to B having more columns than rows, the null space of B changes. Consequently, the generalized singular values of (A, B) change. Thus, it is necessary to perform a $ULLV$ decomposition on the new L_{11} ($(m+1) \times (m+1)$) block to reveal the its rank and update the noise subspace.

Algorithm Updating $ULLVD$ (add a row to B).

1 . Use right rotations to transform y^T in (6.1a) and (6.1b) into $(\eta, 0, \ldots, 0)$.

2 . Permute the $(m+1)$-st and the last rows of the above matrix in the two decompositions (6.2a) and (6.2b).

3 . Turn the last row of the third matrix in (6.3a) and (6.3b) into e_1 using a sequence of column permutations.

4 . Use a left rotation on the first and last rows of the third matrix in (6.4a) and (6.4b) to zero out the last row.

5 . Perform a $ULLV$ decomposition on the new L_{11} matrix to reveal its rank. □

$$
\begin{array}{cccccc}
 & & \downarrow & \downarrow & & \\
\times & 0 & 0 & 0 & 0 & 0 \\
\times & \times & 0 & 0 & 0 & 0 \\
\times & \times & \times & 0 & 0 & 0 \\
\times & \times & \times & \times & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 0
\end{array}
\Longrightarrow
\begin{array}{c}
\\ \\ \rightarrow \\ \rightarrow \\ \\ \\ \\
\end{array}
\begin{array}{cccccc}
\times & 0 & 0 & 0 & 0 & 0 \\
\times & \times & 0 & 0 & 0 & 0 \\
\times & \times & 0 & \times & 0 & 0 \\
\times & \times & \times & \times & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 0
\end{array}
\Longrightarrow
\begin{array}{cccccc}
\times & 0 & 0 & 0 & 0 & 0 \\
\times & \times & 0 & 0 & 0 & 0 \\
\times & \times & \times & 0 & 0 & 0 \\
\times & \times & \times & \times & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 0
\end{array}
$$

Figure 6.1. A permutation, indicated by two down arrows, is followed by a left rotation, indicated by two right arrows, to restore the triangular structure.

$$
\begin{array}{ccccccc}
 & & \downarrow & \downarrow & & & \\
\times & 0 & 0 & 0 & 0 & 0 & 0 \\
\times & \times & 0 & 0 & 0 & 0 & 0 \\
\times & \times & \times & 0 & 0 & 0 & 0 \\
\times & \times & \times & 0 & 0 & 0 & \times \\
\times & \times & \times & 0 & \times & 0 & \times \\
\times & \times & \times & 0 & \times & \times & \times
\end{array}
\Longrightarrow
\begin{array}{c}
\\ \\ \rightarrow \\ \rightarrow \\ \\ \\
\end{array}
\begin{array}{ccccccc}
\times & 0 & 0 & 0 & 0 & 0 & 0 \\
\times & \times & 0 & 0 & 0 & 0 & 0 \\
\times & \times & \times & \times & 0 & 0 & 0 \\
\times & \times & \times & \times & 0 & 0 & \times \\
\times & \times & \times & \times & \times & 0 & \times \\
\times & \times & \times & \times & \times & \times & \times
\end{array}
$$

$$
\Longrightarrow
\begin{array}{ccccccc}
\times & 0 & 0 & 0 & 0 & 0 & 0 \\
\times & \times & 0 & 0 & 0 & 0 & 0 \\
\times & \times & \times & 0 & 0 & 0 & \times \\
\times & \times & \times & \times & 0 & 0 & \times \\
\times & \times & \times & \times & \times & 0 & \times \\
\times & \times & \times & \times & \times & \times & \times
\end{array}
$$

Figure 6.2. A rotation propagates through the second matrix in (6.3a).

ACKNOWLEDGEMENTS

S. Qiao was supported in part by the Natural Sciences and Engineering Research Council of Canada under grant OGP0046301 and by the City University of Hong Kong under a Strategic Research Grant project number 7000611.

REFERENCES

[1] Golub, G.H. and C.F. Van Loan (1989). *Matrix Computations*, 2nd Ed., The Johns Hopkins University Press.

[2] Luk, F.T. and S. Qiao (1994). A new matrix decomposition for signal processing, *Automatica*, Vol. 30, No. 1, pp. 39-43.

[3] Schmidt, R.O. (1981). *A Signal Subspace Approach to Multiple Emitter Location and Spectral Estimation*, Ph.D. dissertation, Stanford University, Stanford, CA.

[4] Schmidt, R.O. (1986). Multiple emitter location and signal parameter estimation, *IEEE Trans. on Antennas and Propagation*, Vol. AP-34, No. 3, pp. 276-280.

[5] Stewart, G.W. (1993). Updating a rank-revealing ULV decomposition, *SIAM J. Matrix Anal. Appl.*, **14**, pp. 494-499.

[6] Van Loan, C.F. (1989). Matrix Computations in Signal Processing, in *Selected Topics in Signal Processing*, S. Haykin, Editor, Prentice Hall, Englewood Cliffs, NJ, pp. 48-72.