DOWNDATING THE RANK-REVEALING URV DECOMPOSITION

HAESUN PARK* AND LARS ELDÉN†

Abstract. We present an accurate algorithm for downdating a row in the rank-revealing URV decomposition, which was recently introduced by Stewart. By downdating the full rank part and the noise part in two separate steps, the new algorithm can produce accurate results even for ill-conditioned problems. Such problems occur e.g. when the rank of the matrix is decreased as a consequence of the downdate. Other possible generalizations of existing QR decomposition downdating algorithms for the rank-revealing URV downdating are discussed. Numerical test results comparing the performance of these new URV decomposition downdating algorithms in the sliding window method are presented.

Key words. downdating, null space, rank-revealing decomposition, sliding window method, two-sided orthogonal decomposition, URV decomposition

AMS(MOS) subject classifications. 65F20, 65F25

1. Introduction. The singular value decomposition (SVD) is of great theoretical and practical importance [8]. One of its major merits is that it provides the rank of the matrix and a basis for four important spaces including the null space. However, the SVD has the drawback that it is computationally expensive. Especially when the problem is of recursive nature, the SVD requires $\mathcal{O}(n^3)$ flops for a matrix of order n even for a simple update such as adding a new row. Thus, algorithms that utilize the existing results for incorporating changes in data are desired. Our goals are to perform such modifications with as few operations and as little storage requirement as possible and compute the new decomposition for rank deficient matrices to obtain accurate results.

Recently, Stewart [18, 17] introduced two-sided orthogonal decompositions, called the rank-revealing URV and ULV decompositions (RR URVD and RR ULVD), which are effective in exhibiting the rank and the basis for the null space, and can be updated in $\mathcal{O}(n^2)$ flops. They are compromises between the SVD and a QR decomposition with some of the virtues of both.

Given a matrix $X \in \mathbf{R}^{p\times n}$, where $p \geq n$, we say that it has numerical rank r, if its singular values satisfy

$$\sigma_1 \geq \cdots \geq \sigma_r > \sigma_{r+1} \geq \cdots \geq \sigma_n,$$

where σ_r is *large* compared to σ_{r+1}. Then there exist orthogonal matrices $U \in \mathbf{R}^{p\times p}$ and $V \in \mathbf{R}^{n\times n}$ such that,

$$U^T X V = \begin{pmatrix} T \\ 0 \end{pmatrix} = \begin{pmatrix} R & F \\ 0 & G \\ 0 & 0 \end{pmatrix} \tag{1.1}$$

* Computer Science Department, University of Minnesota, Minneapolis, MN 55455, U.S.A. (e-mail: hpark@cs.umn.edu). The work of this author was supported in part by the National Science Foundation grant CCR-9209726 and by a contract DAAL02-89-C-0038 between the Army Research Office and the University of Minnesota for the Army High Performance Computing Research Center.

† Department of Mathematics, Linköping University, S–581 83 Linköping, Sweden, (e-mail: laeld@math.liu.se).

where $T \in \mathbf{R}^{n\times n}$, $R \in \mathbf{R}^{r\times r}$, and $G \in \mathbf{R}^{(n-r)\times(n-r)}$ are upper triangular, and

$$\sigma_{\min}(R) \approx \sigma_r, \qquad \| F \|_F^2 + \| G \|_F^2 \approx \sigma_{r+1}^2 + \cdots + \sigma_n^2,$$

where $||\cdot||_F$ denotes the Frobenius norm. This is the RR URVD. Given an upper triangular matrix $T \in \mathbf{R}^{n\times n}$, let w be its unit right singular vector of T corresponding to the smallest singular value σ_n. If an orthogonal matrix $Q \in \mathbf{R}^{n\times n}$ makes

$$Q^T w = e_n$$

and we have the QR decomposition of the product TQ

$$TQ = Q^{(1)}T^{(1)},$$

where $Q^{(1)} \in \mathbf{R}^{n\times n}$ is orthogonal and $T^{(1)} \in \mathbf{R}^{n\times n}$ is upper triangular, then

$$\sigma_n = \| Tw \|_2 = \| {Q^{(1)}}^T TQQ^T w \|_2 = \| T^{(1)} e_n \|_2 .$$

($||\cdot||_2$ denotes the Euclidean vector norm). Thus the last column of $T^{(1)}$ is small. This is called a deflation, and by applying the deflation repeatedly, the RR URVD for T can be computed [18, 17]. When the right orthogonal matrix $V = (V_1\ V_2)$ is partitioned according to the rank, $V_1 \in \mathbf{R}^{n\times r}$ and $V_2 \in \mathbf{R}^{n\times(n-r)}$, then V_2 is an orthogonal basis for the null space of X. Note that the SVD is a special case of the RR URVD.

For recursive problems, there are two common ways for incorporating changes in data, which are the sliding rectangular window method and the exponential window method. For updating the SVD and the RR URVD after a new data row is attached, see [18, 17]. For phasing out the old data, one or more rows are deleted explicitly from X in the sliding rectangular window method, and in the exponential window method, a value β, $0 < \beta < 1$, which is called a *forgetting factor*, is multiplied to existing rows to damp out the effect of the old information. After an update in the exponential window method, the numerical rank can increase, decrease, or stay the same. In particular, since the effect of old data is only gradually phased out, it is likely that there occur situations, where the numerical rank is not well determined, and thus the rank decisions are difficult. In such situations the choice of the forgetting factor and the tolerance used in the decision of the numerical rank becomes critical, especially since these quantities are closely related to each other.

The sliding window method can track the change in the information statistics more accurately than the forgetting factor method when there is an abrupt change in data such as when signals are turned on and off, or outliers are removed [3]. Stewart's original presentation of the URV decomposition was in the context of using an exponential window method in recursive least squares. This paper fills the gap by providing the details for using the URV decomposition in the sliding window setting, making the URV decomposition a complete tool that can handle both types of recursive problems efficiently.

An advantage of the sliding window method is the a priori information on the rank after the modification: mathematically, after adding a row, the rank can only stay the same or increase by one, and, after deleting a row, the rank can only stay the same or decrease by one. Thus, the indefinite steps of deflation in using the forgetting factor that results from not having any similar a priori information on the rank of the modified matrix can be eliminated.

The methods used in this paper are partly based on the algorithms for downdating described in [6], in particular a hybrid between the LINPACK and the CSNE downdating algorithms. It turns out to be necessary to use such accurate algorithms here, since the downdates are often very ill-conditioned when the numerical rank decreases.

The rest of this paper is organized as follows. In Section 2, we discuss the relation between downdating two-sided orthogonal decompositions and the QR decomposition. Then we present a two-step procedure for downdating the QR decomposition. In Section 3, we discuss the rank revealing aspect of URVD downdating and present a new algorithm for downdating the RR URVD based on the two-step QR decomposition downdating algorithm together with other generalizations of QR decomposition downdating algorithms to the RR URVD downdating. Section 4 contains numerical test results comparing the accuracy of these algorithms in deciding rank and a basis for the null space.

There is a vast literature on algorithms in the area of subspace-based signal processing. The papers [1, 4, 19] deal with the problem of rank determination in recursive computations. A Lanczos procedure is used in [19] to find the signal subspace. The algorithm in [4] is based on updating and downdating of a rank-revealing QR decomposition. An application of the exponential window method for the URV decomposition is described in [1].

2. Downdating a two-sided orthogonal decomposition. In this section we first show that the problem of downdating a two-sided orthogonal decomposition is related to that of downdating a QR decomposition. Then we present a two-step procedure for downdating a QR decomposition. This procedure plays an important role in our downdating algorithm for the RR URVD which is presented in the next section. We also introduce a downdating algorithm based on a reduction to a simpler problem.

One important requirement in an algorithm for downdating the rank-revealing URV decomposition is that it must not destroy the "large-small" structure and that it shall reveal the rank after the downdating. In this section, we leave the rank-revealing aspects aside. In the next section, we will show that the two-step procedure combined with condition number estimating methods can fulfill this requirement. The motivation for the two-step procedure is that we separate the downdating of T into two parts to take advantage of the fact that the matrix R is well-conditioned even if T may be ill-conditioned. In a rank-revealing context, since the matrices F and G are considered as "noise", it is undesirable to allow them to destroy the whole downdating procedure due to ill-conditioning.

2.1. A two-step procedure for downdating a QR decomposition. Suppose a two-sided orthogonal decomposition for $X \in \mathbf{R}^{p\times n}$

$$X = U \begin{pmatrix} T \\ 0 \end{pmatrix} V^T = U \begin{pmatrix} R & F \\ 0 & G \\ 0 & 0 \end{pmatrix} V^T \tag{2.1}$$

is given, where U and V are orthogonal, $R \in \mathbf{R}^{r\times r}$ and $G \in \mathbf{R}^{(n-r)\times(n-r)}$ for some $r \leq n$ are upper triangular. We assume that the matrix V is stored, since it is needed in many applications, e.g. subspace-based signal processing and applications that require least squares solutions. However, if $p >> n$ then the extra storage for U may be prohibitive, thus we assume that U is *not stored*. A QR decomposition of X_v defined as

$$X_v \equiv XV = \begin{pmatrix} z^T \\ \tilde{X} \end{pmatrix} V = \begin{pmatrix} z_v^T \\ \tilde{X}_v \end{pmatrix}, \tag{2.2}$$

is

$$X_v = U \begin{pmatrix} T \\ 0 \end{pmatrix}.$$

The QR decomposition downdating problem for X_v

$$X_v = XV = \begin{pmatrix} z_{v1}^T & z_{v2}^T \\ \tilde{X}V_1 & \tilde{X}V_2 \end{pmatrix} = U \begin{pmatrix} R & F \\ 0 & G \\ 0 & 0 \end{pmatrix}, \quad V = (\, V_1 \quad V_2\,),$$

where $z_v^T = (\, z_{v1}^T \quad z_{v2}^T\,)^T$ with $z_{v1} \in \mathbf{R}^{r\times 1}$ and $V_1 \in \mathbf{R}^{n\times r}$, can be formulated as a problem of finding the upper triangular matrix

$$\tilde{T} = \begin{pmatrix} \tilde{R} & \tilde{F} \\ 0 & \tilde{G} \end{pmatrix},$$

given the upper triangular matrix T and the row to be downdated, z_v^T, so that

$$\bar{J}^T \begin{pmatrix} z_{v1}^T & z_{v2}^T \\ \tilde{R} & \tilde{F} \\ 0 & \tilde{G} \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ R & F \\ 0 & G \end{pmatrix},$$

is satisfied for some orthogonal matrix $\bar{J}$. Note that R is the upper triangular factor in the QR decomposition of XV_1. Similarly, $\tilde{R}$ is the upper triangular factor in the QR decomposition of $\tilde{X}V_1$.

Now, we show how we can perform the downdating of the QR decomposition of X_v in two steps based on the partitioning (2.1) of T. This is essential in the new downdating algorithm for the rank-revealing URV decomposition presented in the next section. In obtaining $\tilde{T}$ from T, we can first find $\tilde{R}$ such that

$$\bar{J}_1^T \begin{pmatrix} z_{v1}^T \\ \tilde{R} \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ R \\ 0 \end{pmatrix}, \tag{2.3}$$

for some orthogonal matrix $\bar{J}_1$. This is a standard downdating problem, which can be solved by hyperbolic transformations [2], LINPACK [14, 15], CSNE, or hybrid algorithms [6], for example. With the matrix $\bar{J}_1$ from (2.3), we have the following relation

$$\bar{J}_1^T \begin{pmatrix} z_{v1}^T & z_{v2}^T \\ \tilde{R} & \tilde{F} \\ 0 & \tilde{G} \end{pmatrix} = \begin{pmatrix} 0 & h^T \\ R & F \\ 0 & \tilde{G} \end{pmatrix}, \tag{2.4}$$

for some vector $h \in \mathbf{R}^{(n-r)\times 1}$. The computation of $\tilde{F}$ and h will be discussed at the end of this subsection. Assuming that they are known, we have only to find $\tilde{G}$, such that

$$\bar{J}_2^T \begin{pmatrix} h^T \\ F \\ \tilde{G} \end{pmatrix} = \begin{pmatrix} 0 \\ F \\ G \end{pmatrix}, \tag{2.5}$$

for some orthogonal matrix $\bar{J}_2$. Combining (2.4) and (2.5), we can complete the downdating of the QR decomposition since

$$\bar{J}_2^T \bar{J}_1^T \begin{pmatrix} z_{v1}^T & z_{v2}^T \\ \tilde{R} & \tilde{F} \\ 0 & \tilde{G} \end{pmatrix} = \bar{J}_2^T \begin{pmatrix} 0 & h^T \\ R & F \\ 0 & \tilde{G} \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ R & F \\ 0 & G \end{pmatrix}. \tag{2.6}$$

The two-step procedure is summarized as follows.

Two-step procedure for downdating a 2×2 block QR decomposition
Given the upper triangular factor $\begin{pmatrix} R & F \\ 0 & G \end{pmatrix}$ in the QR decomposition of a matrix, whose first row is $z_v^T = (\, z_{v1}^T \quad z_{v2}^T \,)$, find the downdated triangular factor $\begin{pmatrix} \tilde{R} & \tilde{F} \\ 0 & \tilde{G} \end{pmatrix}$ after deleting z_v^T.

I. Downdate the first part :

(a) Find $\tilde{R}$ such that

$$J_1^T \begin{pmatrix} z_{v1}^T \\ \tilde{R} \end{pmatrix} = \begin{pmatrix} 0 \\ R \end{pmatrix}$$

for some orthogonal matrix J_1.

(b) Determine $\tilde{F}$ and h such that

$$J_1^T \begin{pmatrix} z_{v2}^T \\ \tilde{F} \end{pmatrix} = \begin{pmatrix} h^T \\ F \end{pmatrix}.$$

II. Downdate the second part :

Find $\tilde{G}$ such that

$$J_2^T \begin{pmatrix} h^T \\ \tilde{G} \end{pmatrix} = \begin{pmatrix} 0 \\ G \end{pmatrix},$$

for some orthogonal matrix J_2.

We now discuss Step I(b). After Step I(a), we know $\tilde{R}$ and an orthogonal matrix J_1, such that

$$J_1^T \begin{pmatrix} z_{v1}^T \\ \tilde{R} \end{pmatrix} = \begin{pmatrix} 0 \\ R \end{pmatrix}.$$

Normally J_1 is chosen as a product of plane rotations,

$$J_1^T = P_r \cdots P_2 P_1, \tag{2.7}$$

where P_i is a rotation in the $(1, i+1)$ plane that annihilates the ith element in z_{v1}^T. Specifically, in Step I(b), the first rotation P_1 affects the rows 1 and 2 of $\begin{pmatrix} z_{v2}^T \\ \tilde{F} \end{pmatrix}$, which can be written as

$$P_1(1,2) \begin{pmatrix} z_{v2}^T \\ \tilde{f}_1^T \end{pmatrix} = \begin{pmatrix} h_1^T \\ f_1^T \end{pmatrix}, \quad P_1(1,2) = \begin{pmatrix} c_1 & s_1 \\ -s_1 & c_1 \end{pmatrix}, \tag{2.8}$$

where $c_1 = \cos(\theta_1)$ and $s_1 = \sin(\theta_1)$ for some angle θ_1, the vectors f_1^T and $\tilde{f}_1^T$ denote the first rows of F and $\tilde{F}$, respectively, and $\tilde{f}_1^T$ and h_1^T are unknown. We use the notation $P_1(i,j)$ for the 2×2 matrix, whose elements are the $(i,i), (i,j), (j,i)$ and (j,j) elements of P_1. Rearranging (2.8), we obtain

$$H_1(1,2)\begin{pmatrix} h_1^T \\ \tilde{f}_1^T \end{pmatrix} = \begin{pmatrix} z_{v2}^T \\ f_1^T \end{pmatrix}, \quad H_1(1,2) = \begin{pmatrix} 1/c_1 & s_1/c_1 \\ s_1/c_1 & 1/c_1 \end{pmatrix},$$

from which we have

$$\begin{pmatrix} h_1^T \\ \tilde{f}_1^T \end{pmatrix} = H_1(1,2)^{-1}\begin{pmatrix} z_{v2}^T \\ f_1^T \end{pmatrix} = \begin{pmatrix} 1/c_1 & -s_1/c_1 \\ -s_1/c_1 & 1/c_1 \end{pmatrix}\begin{pmatrix} z_{v2}^T \\ f_1^T \end{pmatrix},$$

where $H_1(1,2)$ can be identified as a hyperbolic rotation since $1/c_1 = 1/\cos(\theta_1) = \cosh(\eta_1)$ and $\sin(\theta_1)/\cos(\theta_1) = \sinh(\eta_1)$ for some η_1 [2]. We can continue in a similar way and compute $\tilde{F}$ and h as shown in the following algorithm.

Algorithm HYPER
Given $F \in \mathbf{R}^{r\times(n-r)}$, $z_{v2} \in \mathbf{R}^{(n-r)\times 1}$, and rotation parameters in vectors $cc \in \mathbf{R}^{r\times 1}$ and $ss \in \mathbf{R}^{r\times 1}$ from downdating transformations for R, compute $\tilde{F}$ and h.

1. Determine elementary transformations

$$H_i^{-1}(1,i+1) = \begin{pmatrix} 1/cc(i) & -ss(i)/cc(i) \\ -ss(i)/cc(i) & 1/cc(i) \end{pmatrix}$$

2. Compute

$$\begin{pmatrix} h^T \\ \tilde{F} \end{pmatrix} = H_r^{-1}\cdots H_1^{-1}\begin{pmatrix} z_{v2}^T \\ F \end{pmatrix}$$

Algorithm HYPER requires about $4r(n-r)$ flops (1 flop is 1 multiplication and 1 addition). The matrix V does not change since the transformations are applied from the left side only.

2.2. Reduction to a simpler downdating problem. From Section 2.1, we know that if we have the downdated QR decomposition of X_v after deleting z_v^T,

$$\tilde{X}_v = \tilde{U}\begin{pmatrix} \tilde{T} \\ 0 \end{pmatrix},$$

where $\tilde{U}$ is orthogonal and $\tilde{T}$ is upper triangular, then we have the downdated two-sided decomposition of X after deleting z^T, since

$$\tilde{X} = \tilde{U}\begin{pmatrix} \tilde{T} \\ 0 \end{pmatrix} V^T.$$

Similarly, if we have a two-sided orthogonal decomposition for $\tilde{X}_v$,

$$\tilde{X}_v = \hat{U}\begin{pmatrix} \hat{T} \\ 0 \end{pmatrix} V_0^T,$$

for some orthogonal matrices $\hat{U}$ and V_0, and a triangular matrix $\hat{T}$, then

$$\tilde{X} = \hat{U}\begin{pmatrix}\hat{T}\\ 0\end{pmatrix}\hat{V}^T, \quad \hat{V} = VV_0$$

gives a downdated two-sided orthogonal decomposition of $\tilde{X}$.

In the two-sided orthogonal decomposition, we can transform the downdating problem into a simpler one by reducing z_{v1} to κe_r, where $\kappa = ||z_{v1}||_2$, by a procedure analogous to one in [17]. This can be done without changing the "large-small" structure of the upper triangular matrix, using a sequence of Givens rotations. We can find matrices $V_0' = Y_1 \cdots Y_{r-1}$ and $U_0' = W_1 \cdots W_{r-1}$, where $Y_i \in \mathbf{R}^{r\times r}$ and $W_i \in \mathbf{R}^{r\times r}$ are rotation matrices, such that $z_{v1}^T V_0' = \kappa e_r^T$, and $R^{(1)} = U_0'^T R V_0'$ is upper triangular. Computation of $R^{(1)}$ requires $O(r^2)$ operations.

Defining

$$V^{(1)} = V\begin{pmatrix}V_0' & 0\\ 0 & I_{n-r}\end{pmatrix}, \qquad U^{(1)} = U\begin{pmatrix}U_0' & 0\\ 0 & I_{p-r}\end{pmatrix},$$

and $F^{(1)} = U_0'^T F$, we now have

$$XV^{(1)} = \begin{pmatrix}\kappa e_r^T & z_{v2}^T\\ \tilde{X}V_1^{(1)} & \tilde{X}V_2^{(1)}\end{pmatrix} = U^{(1)}\begin{pmatrix}R^{(1)} & F^{(1)}\\ 0 & G\\ 0 & 0\end{pmatrix}, \tag{2.9}$$

where $V^{(1)} = (V_1^{(1)}\ V_2^{(1)})$ and $V_1^{(1)} \in \mathbf{R}^{n\times r}$, and (2.9) is another two-sided orthogonal decomposition of the matrix X. Accordingly, (2.9) gives a QR decomposition of the matrix $XV^{(1)}$ for which the first row to be downdated has a simpler form than that in (2.2). Furthermore, the norm of each submatrix of T has been maintained, i.e., $||R||_F = ||R^{(1)}||_F$ and $||F||_F = ||F^{(1)}||_F$.

In Step I(a) of the two-step procedure of Section 2, we now want to find an upper triangular $\tilde{R}$, such that

$$J_1^T\begin{pmatrix}\kappa e_r^T\\ \tilde{R}\end{pmatrix} = \begin{pmatrix}0\\ R^{(1)}\end{pmatrix}, \tag{2.10}$$

for some orthogonal matrix J_1. Writing the matrix $R^{(1)}$ in the form

$$R^{(1)} = \begin{pmatrix}R_1 & s\\ 0 & \rho\end{pmatrix},$$

where $R_1 \in \mathbf{R}^{(r-1)\times(r-1)}$, and $s \in \mathbf{R}^{(r-1)\times 1}$, we see that the downdated matrix $\tilde{R}$ differs from $R^{(1)}$ only in the (r,r) element, and it is given by

$$\tilde{R} = \begin{pmatrix}R_1 & s\\ 0 & \tilde{\rho}\end{pmatrix}, \quad \tilde{\rho} = \sqrt{\rho^2 - \kappa^2}. \tag{2.11}$$

Now Step I(a) in the two-step procedure consists of computing $\tilde{\rho}$ as in (2.11). Since the row to be downdated has already been transformed to κe_r^T, we only need to multiply by one hyperbolic rotation in Step I(b) to obtain h^T and the last row of $\tilde{F}^{(1)}$.

This procedure can be used also when the downdating fails due to $\rho^2 - \kappa^2 < 0$ in floating point arithmetic. In this case we can simply put $\tilde{\rho}$ equal to zero. In the next section we will also use it for downdating G in the two-step algorithm. We summarize the algorithm as follows.

Algorithm REDUCTION
Given $R \in \mathbf{R}^{r\times r}$ and $v \in \mathbf{R}^{r\times 1}$, this algorithm computes the downdated matrix $\tilde{R} \in \mathbf{R}^{r\times r}$, by using two-sided transformations on R to reduce the vector v into a simpler form keeping the triangular structure of R.

1. Determine plane rotations Y_i in the plane $(i, i+1)$, $1 \le i \le r-1$, such that

$$\kappa e_r^T = v^T Y_1 \cdots Y_{r-1}$$

   where $\kappa = ||v||_2$.
2. Determine plane rotations W_i in the plane $(i, i+1)$, $1 \le i \le r-1$, such that

$$\tilde{R} := W_{r-1}^T \cdots W_1^T R Y_1 \cdots Y_{r-1}$$

   is upper triangular.
   Compute $\tilde{\alpha} := \tilde{R}(r,r)^2 - \kappa^2$
   If $\tilde{\alpha} > 0$ then
   $\tilde{R}(r,r) = \sqrt{\tilde{\alpha}}$
   else
   $\tilde{R}(r,r) = 0$
   end if
   Save the information for Y_i and W_i if necessary.

This algorithm requires $4r^2$ flops. When algorithm REDUCTION is used in Step I(a) of the two-step downdating, then the matrix F must also be modified with W_i's ($4r(n-r)$ flops). Also the matrix V has to be modified due to the right side transformations $Y_1 \cdots Y_{r-1}$ ($4nr$ flops). Altogether, this adds up to $8nr$ flops. Similar operation counts are obtained when this algorithm is used in Step II. As a result of reducing the vector v into κe_r^T, the actual downdating occurs only on the last column of the new triangular factor. The information from the plane rotations are saved if necessary.

3. Downdating the rank-revealing URV decomposition. We now extend the results from the previous section and consider downdating the rank-revealing URV decomposition of X

$$X = U \begin{pmatrix} R & F \\ 0 & G \\ 0 & 0 \end{pmatrix} V^T, \tag{3.1}$$

where $R \in \mathbf{R}^{r\times r}$ and G are upper triangular, and the matrices F and G are assumed to be small in the sense that they satisfy

$$\nu = \sqrt{||F||_F^2 + ||G||_F^2} \le tol,$$

for a given value tol. Thus X has numerical rank r.

We have seen in Section 2 that downdating the QR decomposition of XV is closely related to downdating the two-sided orthogonal decomposition of X. In this section, we incorporate the aspects of numerical rank decisions, in particular. The problem of downdating the RR URVD (3.1) can be formulated as follows:

Problem : Downdating of RR URVD Given $\begin{pmatrix} R & F \\ 0 & G \end{pmatrix}$, V, and $r = rank(R)$ in the RR URVD of

$$X = \begin{pmatrix} z^T \\ \tilde{X} \end{pmatrix} = U \begin{pmatrix} R & F \\ 0 & G \\ 0 & 0 \end{pmatrix} V^T,$$

find the downdated matrices $\bar{V}$, $\bar{R}$, $\bar{F}$, and $\bar{G}$ and $\bar{r} = rank(\bar{R})$ such that

$$\tilde{X} = \bar{U} \begin{pmatrix} \bar{R} & \bar{F} \\ 0 & \bar{G} \\ 0 & 0 \end{pmatrix} \bar{V}^T$$

for some orthogonal matrix $\bar{U}$.

When the numerical rank of $\bar{T}$ is reduced by one as a result of downdating, then $\bar{r}$ is $r-1$, and otherwise it is equal to r. Thus, after downdating, $\bar{T}$ not only is upper triangular but also reveals the new rank, i.e.,

$$\bar{\nu} = \sqrt{||\bar{F}||_F^2 + ||\bar{G}||_F^2} \le tol.$$

Note the difference in notations from those in Section 2: $\tilde{}$ is used to denote the downdated matrices before the rank decision is made and $\bar{}$ is used to denote those after the rank decision is made on downdated matrices. Thus the order of $\bar{R}$ is one less than that of $\tilde{R}$ when the rank of R is reduced by one as a result of downdating.

3.1. Algorithms for two-step downdating. The downdating algorithm based on hyperbolic rotations [2], the LINPACK [14, 15], hybrid, and CSNE [6] downdating algorithms can be used for the two-step downdating. In fact, when hyperbolic rotations are used, it is not necessary to divide the downdating of X_v into two steps, as the same procedure can be applied throughout. Below we summarize the LINPACK and LINPACK/CSNE hybrid algorithms. For more details, see [6, 14, 15].

When the LINPACK algorithm is used for Step I(a) in the two-step downdating, we first solve the triangular system $R^T q_1 = z_{v1}$ and compute the plane rotations $P_1, \cdots, P_r$ so that

$$P_1^T \cdots P_r^T \begin{pmatrix} \gamma \\ q_1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \tag{3.2}$$

where $\gamma = \sqrt{1 - ||q_1||^2}$, which also gives

$$P_1^T \cdots P_r^T \begin{pmatrix} 0 \\ R \end{pmatrix} = \begin{pmatrix} z_{v1}^T \\ \tilde{R} \end{pmatrix}. \tag{3.3}$$

In (3.2), each P_i^T is a rotation in the $(1, i+1)$ plane and (c_i, s_i) from P_i can be saved to obtain H_i to perform Step I(b).

The equation (3.3) and the discussion of Step I(b) in Section 2.1 show how the LINPACK algorithm and the algorithm based on hyperbolic rotations are related: equivalent transformations are applied, in the first case as Givens rotations, and in the second as hyperbolic rotations (cf. also [11]). The LINPACK algorithm is summarized in the following.

Algorithm LINPACK
Given $R \in \mathbf{R}^{r\times r}$ and $v \in \mathbf{R}^{r\times 1}$, this algorithm computes the downdated matrix $\tilde{R}$ and saves the rotation parameters from downdating transformation in $cc \in \mathbf{R}^{r\times 1}$ and $ss \in \mathbf{R}^{r\times 1}$.

1. Compute q_1 and γ from

$$R^T q_1 = v, \qquad \gamma := (1 - ||q_1||^2)^{1/2}$$

2. Determine plane rotations P_i in the $(1, i+1)$ plane, $1 \le i \le r$, such that

$$\begin{pmatrix} 1 & v^T \\ 0 & \tilde{R} \end{pmatrix} := P_1^T \cdots P_r^T \begin{pmatrix} \gamma & 0 \\ q_1 & R \end{pmatrix}$$

and save

$$cc(i) = c_i, \quad ss(i) = s_i$$

where (c_i, s_i) is from $P_i(1, i+1) = \begin{pmatrix} c_i & s_i \\ -s_i & c_i \end{pmatrix}$, if necessary.

The above algorithm requires about $2.5r^2$ flops. There is no change in the matrix V in the URVD.

The downdating problem in Step I(a) in the two-step procedure is ill-conditioned in two cases: a) *when a row containing an outlier is removed*, and b) *when there is a decrease in rank* (cf.[15, 6]). The ill-conditioning reveals itself in that the quantity $1 - ||q_1||_2^2$ computed in the LINPACK algorithm is small, or even negative in floating point arithmetic. In [6] it was shown that the method of Corrected SemiNormal Equations (CSNE) can be used to recover q_1 and γ more accurately in ill-conditioned cases. Accordingly, the P_i's in (3.2) are also determined more accurately. The accurate results from the CSNE method are obtained due to the fact that the original data matrix is used in the refinement of q_1 and γ. The hybrid method [6] that applies the refinement only when necessary, is an excellent compromise when both computational complexity and accuracy are concerns.

We will refer to the LINPACK, hybrid, and CSNE algorithms as LINPACK-type algorithms since the way the downdating transformations are computed in these methods is similar. The essential difference among LINPACK-type algorithms is in the computation of q_1 and γ. After q_1 and γ are determined, the rest of the algorithm is the same. Thus, each P_i is computed to satisfy (3.2) and the Step I(b) can be performed using H_i's obtained from P_i's. Note that in the first step of downdating, these LINPACK-type algorithms will do well in general in computing q_1 since R is relatively well-conditioned according to the definition of the RR URVD. The hybrid LINPACK/CSNE algorithm is summarized below.

Algorithm LINPACK/CSNE
Given $X \in \mathbf{R}^{p\times n}$, the first r columns of V, which is $V_1 \in \mathbf{R}^{n\times r}$, $R \in \mathbf{R}^{r\times r}$, $v \in \mathbf{R}^{r\times 1}$, and a tolerance $tol1$, this algorithm computes the downdated matrix $\tilde{R}$ and saves the rotation parameters from downdating transformation in $cc \in \mathbf{R}^{r\times 1}$ and $ss \in \mathbf{R}^{r\times 1}$.

1. Compute q_1 and α from

$$R^T q_1 = v, \qquad \alpha := 1 - ||q_1||_2^2.$$

2. If $\alpha > tol1$ then (LINPACK)
   $\gamma := \sqrt{\alpha}$
   else (CSNE)
   Compute y and t from

$$Ry = q_1, \qquad t := e_1 - XV_1 y.$$

   Improve q_1, y and compute γ

$$R^T \delta q_1 = V_1^T X^T t, \qquad q_1 := q_1 + \delta q_1,$$
$$R \delta y = \delta q_1, \qquad t := t - XV_1 \delta y, \qquad \gamma := \|t\|_2.$$

   end if
3. Determine plane rotations P_i in the $(1, i+1)$ plane, $1 \le i \le r$ such that

$$\begin{pmatrix} 1 & v^T \\ 0 & \tilde{R} \end{pmatrix} := P_1^T \cdots P_r^T \begin{pmatrix} \gamma & 0 \\ q_1 & R \end{pmatrix},$$

   and save

$$cc(i) = c_i, \quad ss(i) = s_i,$$

   where (c_i, s_i) is from $P_i(1, i+1) = \begin{pmatrix} c_i & s_i \\ -s_i & c_i \end{pmatrix}$, if necessary.

This algorithm requires $3np + 3nr + 4r^2$ flops if the CSNE branch is taken and $2.5r^2$ flops otherwise. There is no change in V in the URVD. Another alternative for Step I(a) downdating is the reduction algorithm described in Section 2.2.

When G is ill-conditioned or numerically singular, then downdating in Step II will be difficult and the LINPACK-type algorithms will suffer since the downdating transformation relies on the solution of the triangular system where G is the coefficient matrix. The downdating method based on hyperbolic rotations will have similar problems as the LINPACK algorithm. If we use the reduction algorithm in Step II, then we first reduce the vector h^T obtained from Step I(b) into τe_{n-r}^T where $\tau = ||h||_2$ while restoring the triangular structure of G as shown in Section 2.2. Even when G is ill-conditioned or singular, the reduction step will not break down.

3.2. Deflation. In downdating, the numerical rank cannot increase, but can decrease. Therefore, we need to test $\tilde{R}$ obtained from Step I(a) for numerical rank deficiency and deflate it, if necessary. Rank deficiency of a matrix can be detected by estimating its smallest singular value [8, p. 128], [9]. In downdating, there are two cases to consider. After downdating and before deciding the new rank, the new value for ν is

$$\tilde{\nu}^2 = \nu^2 - ||z_{v2}||_2^2, \tag{3.4}$$

which is from

$$\nu^2 = ||F||_F^2 + ||G||_F^2 = ||\tilde{F}||_F^2 + ||\tilde{G}||_F^2 + ||z_{v2}||_2^2 = \tilde{\nu}^2 + ||z_{v2}||_2^2.$$

Suppose

$$\tilde{R} = \begin{pmatrix} \tilde{R}_1 & \tilde{s} \\ 0 & \tilde{\rho} \end{pmatrix}, \tag{3.5}$$

where $\tilde{R}_1 \in \mathbf{R}^{(r-1)\times(r-1)}$. If

$$\sqrt{\tilde{\nu}^2 + ||\tilde{s}||^2 + \tilde{\rho}^2} \leq tol, \tag{3.6}$$

then the numerical rank is decreased as a result of downdating and $\bar{R} = \tilde{R}_1 \in \mathbf{R}^{(r-1)\times(r-1)}$.

If (3.6) is not satisfied, then the rank may decrease or stay the same. To check this, the smallest singular value of $\tilde{R}$ is estimated and the deflation step [17] mentioned in the introduction is applied to transform $\tilde{R}$ to

$$W^T \tilde{R} Y = \begin{pmatrix} \tilde{R}_d & \tilde{s}_d \\ 0 & \tilde{\rho}_d \end{pmatrix}, \tag{3.7}$$

where W and Y are orthogonal, $\tilde{R}_d \in \mathbf{R}^{(r-1)\times(r-1)}$ is upper triangular, and $\hat{\sigma}_d = \sqrt{\tilde{\rho}_d^2 + ||\tilde{s}_d||_2^2}$ is a good approximation of the smallest singular value of $\tilde{R}$. If, after transforming $\tilde{R}$ as in (3.7),

$$\sqrt{\tilde{\nu}^2 + \hat{\sigma}_d^2} \leq tol,$$

then the numerical rank of $\tilde{R}$ is $r-1$, and we can modify $\tilde{F}$ and $\tilde{G}$ accordingly. In this case, the matrices F and V must be updated by W and Y, respectively. Otherwise, the rank after the downdating is the same as before, so $\bar{r} = r$, and $\bar{R} = \tilde{R}$.

We outline the deflation procedure as follows.

Algorithm DEFLATE

Given the downdated matrices $\tilde{R} \in \mathbf{R}^{r\times r}, \tilde{F} \in \mathbf{R}^{r\times(n-r)}$, $\tilde{G} \in \mathbf{R}^{(n-r)\times(n-r)}$, and the orthogonal $\tilde{V} \in \mathbf{R}^{n\times n}$, test for deflation and perform deflation if needed.

1. Compute $\tilde{\nu}^2 = \nu^2 - ||z_{v2}||^2$.

2. $\bar{R} = \tilde{R}$, $\bar{F} = \tilde{F}$, $\bar{G} = \tilde{G}$, $\bar{V} = \tilde{V}$.
   If $\sqrt{\tilde{\nu}^2 + ||\bar{R}(:,r)||_2^2} \leq tol$ then
   $\bar{r} := r-1$, $\bar{R} = \bar{R}(1:r-1, 1:r-1)$, $\bar{\nu} := \sqrt{\tilde{\nu}^2 + ||\bar{R}(:,r)||_2^2}$,
   change dimensions of $\bar{F}$ and $\bar{G}$ accordingly.
   else
   Estimate the smallest singular value, $\hat{\sigma}$ of $\bar{R}$.
   If $\sqrt{\tilde{\nu}^2 + \hat{\sigma}^2} < tol$,
   Transform $\bar{R}$ as in (3.7), $\bar{R} := W^T \bar{R} Y$, and modify $\bar{F}$ and $\bar{V}$.
   $\bar{r} := r-1$, $\bar{R} = \bar{R}(1:r-1, 1:r-1)$, $\bar{\nu} := \sqrt{\tilde{\nu}^2 + \hat{\sigma}^2}$.
   Modify $\bar{F}$ and $\bar{G}$ accordingly.
   end if
   end if

The estimation of the smallest singular value requires αr^2 flops, where α is a small constant [9]. The transformation (3.7) can be done in $8nr$ flops (including the modification of F and V).

3.3. New algorithms. Incorporating the rank decision with the downdating algorithms described in the previous sections, we present three RR URVD downdating algorithms. Various algorithms can be constructed by combining the algorithms presented for Step I(a) and Step II downdating. Our goal is to obtain accurate solutions with less computational work. The first obvious choice is applying the LINPACK algorithm for both Steps I and II. However, LINPACK can fail in Step II (in fact, it can fail also in Step I(a), see the numerical tests in Section 4). Therefore, if the downdating in Step II is very ill-conditioned, or, if it breaks down completely (this can be seen in the LINPACK algorithm in that the quantity, $1 - ||q_1||_2^2$, becomes negative, which, in theory, is positive), then we use REDUCTION algorithm, in which we simply set the last diagonal element of $\tilde{G}$ equal to zero when the downdating breaks down.

Another choice is to use the REDUCTION algorithm in Steps I(a) and II. In this case, only one hyperbolic rotation is needed in Step I(b).

We want our algorithm to be robust when the downdating is ill-conditioned in Step I(a) or Step II, and we want it to be fast at the same time. The LINPACK algorithm is considerably less expensive than the CSNE algorithm, while the latter has much better accuracy than the former [6]. Thus we can use the hybrid LINPACK/CSNE algorithm in Step I(a). In Step II, we prefer the LINPACK algorithm in case the downdating is well-conditioned, and the REDUCTION algorithm otherwise, since the latter is less expensive than the CSNE algorithm, and can also deal with the case when G is exactly singular. This is the third algorithm.

The three algorithms are summarized in Table 1 and the total computational complexity is shown in Table 2. In combining the different algorithms described above into a two-step procedure, and in computing complexity, the following rules should be applied:

1. If R is modified from the left, then the same transformation must be applied to F.
2. If G is modified from the right, then the same transformation must be applied to F.
3. If any part is modified from the right, then V must be modified accordingly.

Step	Algorithm A	Algorithm B	Algorithm C
I(a)	LINPACK	REDUCTION	LINPACK/CSNE
I(b)	HYPER	HYPER	HYPER
II	LINPACK/REDUCTION	REDUCTION	LINPACK/REDUCTION
III	DEFLATE	DEFLATE	DEFLATE

TABLE 3.1
Three algorithms

Step	Algorithm A	Algorithm B	Algorithm C
I(a)	$2.5r^2$	$8nr$	$2.5r^2$ or $3np + 3nr + 4r^2$
I(b)	$4r(n-r)$	$4(n-r)$	$4r(n-r)$
II	$2.5(n-r)^2$ or $4(n-r)(2n-r)$	$4(n-r)(2n-r)$	$2.5(n-r)^2$ or $4(n-r)(2n-r)$
III	αr^2 $(+8nr)$	αr^2 $(+8nr)$	αr^2 $(+8nr)$

TABLE 3.2
Computational complexity (flops)

4. Numerical Experiments. Numerical tests have been performed in Pro-Matlab with IEEE double precision floating point arithmetic to compare the accuracy of the RR URVD downdating algorithms that have been presented. In a sliding window method, the data matrix consists of the p latest rows of an observation matrix. In each step, a new row of observations is updated into the RR URVD, and an existing row of the data matrix is downdated from the decomposition, on a first in, first out basis. The QR decomposition or the SVD of the window matrix was used as a reference in checking the accuracy.

In the plots where signal and noise space errors are shown, the solid line, the dashed line, and the dashed-dotted lines denote results from Algorithm A, Algorithm B, and Algorithm C, respectively. Errors are plotted against the step number in the sliding window method.

The error in the signal part was taken to be the 2-norm of the difference between the recursively computed matrix R and the R factor from the QR decomposition of the window matrix multiplied by the first r columns of V, where r is the computed rank from the corresponding algorithm.

The basis of the numerical null-space given by the last $n-r$ columns of V was compared to the null-space obtained from the SVD of the window matrix. The largest principal angle between these two null space bases was computed as $\arccos(\nu_{min})$, where ν_{min} is the smallest singular value of $V_0^T Z_0$, and V_0 and Z_0 are the orthonormal null space bases from the computed V and the SVD, respectively, see [5] and [8, p. 585]. From [10], the imprecision in the null-space due to the non-zero block F in the RR URVD is essentially $||F||_F^2/\sigma_r(R)$. This level of uncertainty depends on the details of the deflation procedure and thus it may vary for different algorithms.

According to the perturbation analysis result of Pan[12] (see also [15, 7]), the downdating condition number is $\kappa_{down} = \kappa^2(R)/(1-||q_1||_2^2)$, where $\kappa(R) = \|R\|_2 \, \|R^{-1}\|_2$ is the condition number of the matrix to be downdated. In each plot where the condition numbers for downdating are shown, the solid line and the dashed line represents $1/(1-||q_1||_2^2)$, and $\kappa(R)$, respectively. When $1-||q_1||_2 < 0$ numerically, then $1/(1-||q_1||_2^2)$ is set to be 10^{20} in the plot. In Algorithm C, Step I(a), the CSNE branch was taken when $\kappa(R) > 10^4$, or $1/(1-||q_1||_2^2) > 5$. In Algorithms A and C, Step II, the Linpack branch was taken if $\kappa(G) < 6 \cdot 10^6$, and $1/(1-||q_1||_2^2) \leq 5$. It is possible to estimate the condition numbers of the triangular matrices after updating/downdating in $O(n)$ operations with the adaptive methods such as ACE [13].

In the graph where rank estimates are presented, the $\times$ signs indicate where the CSNE branch was used in Algorithm C and $+$ signs indicate where the downdating in the REDUCTION algorithm in Step I(a) was made by setting the diagonal element equal to zero.

Test I. A random matrix $A \in \mathbf{R}^{140\times 8}$ was constructed with elements taken from a uniform distribution in $(0,1)$. In order to vary the numerical ranks of the window matrices, certain parts of the matrix were multiplied by $\delta = 10^{-4}$. Then the matrix was multiplied by a random, orthogonal 8×8 matrix from the right. Finally, an outlier equal to 10^3 was added in position (8,4). The window size p was 12. The tolerance value for determining the numerical rank was 10δ. The results are shown in Figures 1 and 2.

Algorithms A and B lost accuracy significantly in signal space estimation after the outlier was deleted. Then the accuracy was not recovered even after the downdating problems became well-conditioned. This phenomenon is consistent with what was observed in the LINPACK algorithm for QR decomposition downdating [6].

All three algorithms gave the correct estimate of numerical ranks. Note that the first rank reduction happened when the outlier was removed. The noise space error was below the level of uncertainty given by the data for all three algorithms. They produced almost identical results. In noise space downdating, the numerical value for $1 - ||q_1||_2^2$ was often negative toward the last steps.

Test II. A random matrix $A \in \mathbf{R}^{74 \times 9}$ was constructed with ill-conditioned diagonal blocks and zero blocks. Then the matrix was multiplied by random orthogonal matrices from the left and right sides. The window size p was 13. The tolerance value for determining the numerical rank was 30δ. The results are shown in Figures 3 and 4.

Algorithm A failed when κ_{down} became negative in Step I(a). Algorithms B and C produced comparable results although Algorithm C was consistently better. The error in the noise subspace was again of the magnitude that one would expect from the fact that the noise level in the data and the tolerance. For a sequence of steps, the plot of the noise error shows an error equal to zero since the window matrix had full column rank, and thus there was no non-trivial null-space.

All three algorithms gave the correct estimate of numerical ranks. Note that the CSNE branch was heavily used when the rank keeps decreasing in Algorithm C.

We have run several other tests. The results reported here are typical. However, there were a few cases, with singular G, when the performance of the three algorithms was less satisfactory. For such problems, we have developed a method, which is based on the observation that G is the R factor of the QR decomposition of the residual matrix for the least squares problem $\min_W \|XV_1W - XV_2\|_F$, where $V = (V_1\ V_2)$, $V_1 \in \mathbf{R}^{n \times r}$. The solution of this problem is obtained from the linear system $RW = F$, and it can be refined using CSNE. Preliminary tests have shown promising results, and, although this method is considerably more costly than Algorithm Reduction, it may be an alternative for handling extreme cases with almost singular G. Also Algorithms A and B seem to be more sensitive to the choice of the tolerance value for the rank decision. The actual applications should provide helpful information for the tolerances. Further work is needed to make the algorithms more robust.

5. Conclusion. In this paper we have introduced a two-step procedure for downdating the rank-revealing URV decomposition, where the downdating of the signal and noise parts are performed separately. This enables us to obtain accurate results using the LINPACK/CSNE hybrid algorithm for the ill-conditioned downdates, which occur when the numerical rank is decreased.

Three different algorithms based on the two-step procedure have been described. The numerical tests indicate that it is necessary to use the more sophisticated variants to handle ill-conditioned downdates.

In signal processing applications, where signals are turned on and off (thus the numerical ranks of the window matrices vary), the sliding window method has been reported to be highly sensitive to round-off errors [3], and therefore the exponential windowing method has been preferred. However, our preliminary tests indicate that the new algorithms, and Algorithm C in particular, can give accurate results for the ill-conditioned downdates that occur when rank decreases. We expect that the new algorithms can be used successfully for such applications. Further research is needed to investigate their applicability to real problems in signal processing and to study the choice of the tolerance for rank decision, see [16].

REFERENCES

[1] G. Adams, M. F. Griffin, and G. W. Stewart, *Direction-of-arrival estimation using the rank-revealing URV decomposition*, Tech. Report CS-TR-2640, Dept. of Computer Science, Univ. of Maryland, March 1991.

[2] S. T. Alexander, C.-T. Pan, and R. J. Plemmons, *Analysis of a recursive least squares hyperbolic rotation algorithm for signal processing*, Lin. Alg. Appl., 98 (1988), pp. 3–40.

[3] M. G. Bellanger, *The family of fast least squares algorithms for adaptive filtering*, in Mathematics in Signal Processing, J. McWhirter, ed., Clarendon Press, Oxford, 1990, pp. 415–434.

[4] C. H. Bischof and G. M. Schroff, *On updating signal subspaces*, IEEE Trans. Signal Proc., 40 (1992), pp. 96–105.

[5] A. Björck and G. H. Golub, *Numerical methods for computing angles between subspaces*, Math. Comp., 27 (1973), pp. 579–594.

[6] Å. Björck, H. Park, and L. Eldén, *Accurate downdating of least squares solutions*, SIAM J. Matrix Anal. Appl., to appear, (1994).

[7] L. Eldén and H. Park, *Perturbation analysis for block downdating of a Cholesky decomposition*, Numersiche Mathematik, to appear, (1994).

[8] G. H. Golub and C. F. Van Loan, *Matrix Computations. 2nd ed.*, Johns Hopkins Press, Baltimore, MD., 1989.

[9] N. J. Higham, *Experience with a matrix norm estimator*, SIAM J. Sci. Stat. Comput., 11 (1990), pp. 804–809.

[10] R. Mathias and G. W. Stewart, *A block QR algorithm and the singular value decomposition*, Linear Alg. Appl., 182 (1993), pp. 91–100.

[11] C.-T. Pan, *A modification to the Linpack downdating algorithm*, BIT, 30 (1990), pp. 707–722.

[12] ———, *A perturbation analysis of the problem of downdating a Cholesky factorization*, Lin. Alg. Appl., 183 (1993), pp. 103–116.

[13] D. J. Pierce and R. J. Plemmons, *Fast adaptive condition estimation*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 274–291.

[14] M. A. Saunders, *Large-scale linear programming using the Cholesky factorization*, Tech. Report Technical Report CS252, Computer Science Department, Stanford University, 1972.

[15] G. W. Stewart, *The effects of rounding error on an algorithm for downdating a Cholesky factorization*, Journal of the Institute for Mathematics and Applications, 23 (1979), pp. 203–213.

[16] G. W. Stewart, *Determining rank in the presence of error*, Tech. Report CS-TR-2972, Dept. of Computer Science, Univ. of Maryland, 1992.

[17] G. W. Stewart, *An updating algorithm for subspace tracking*, IEEE Trans. Signal Proc., 40 (1992), pp. 1535–1541.

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

[19] G. Xu, H. Zha, G. H. Golub, and T. Kailath, *Fast and robust algorithms for updating signal subspaces*, Tech. report, Information Systems Lab. Stanford University, 1991.

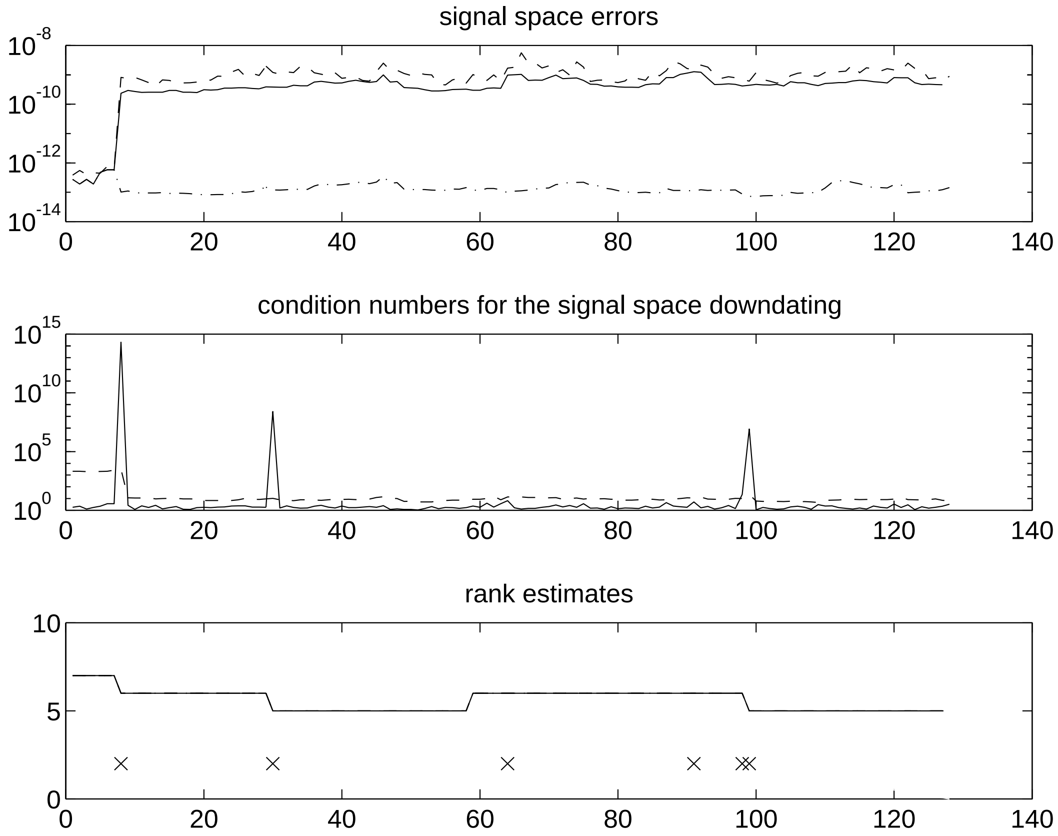

FIG. 5.1. *Test I. Signal space errors, downdating conditioning, and rank estimates for* 12×8 *window matrices.*

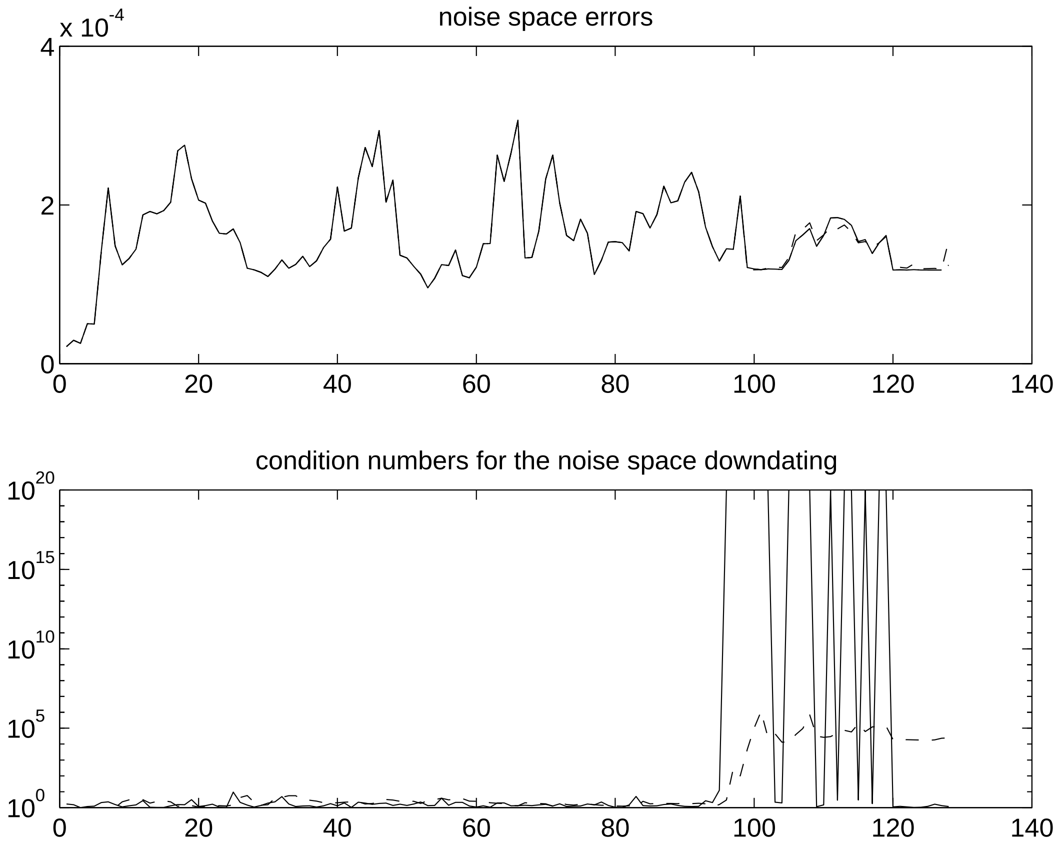

FIG. 5.2. *Test I. Noise space error and downdating conditioning.*

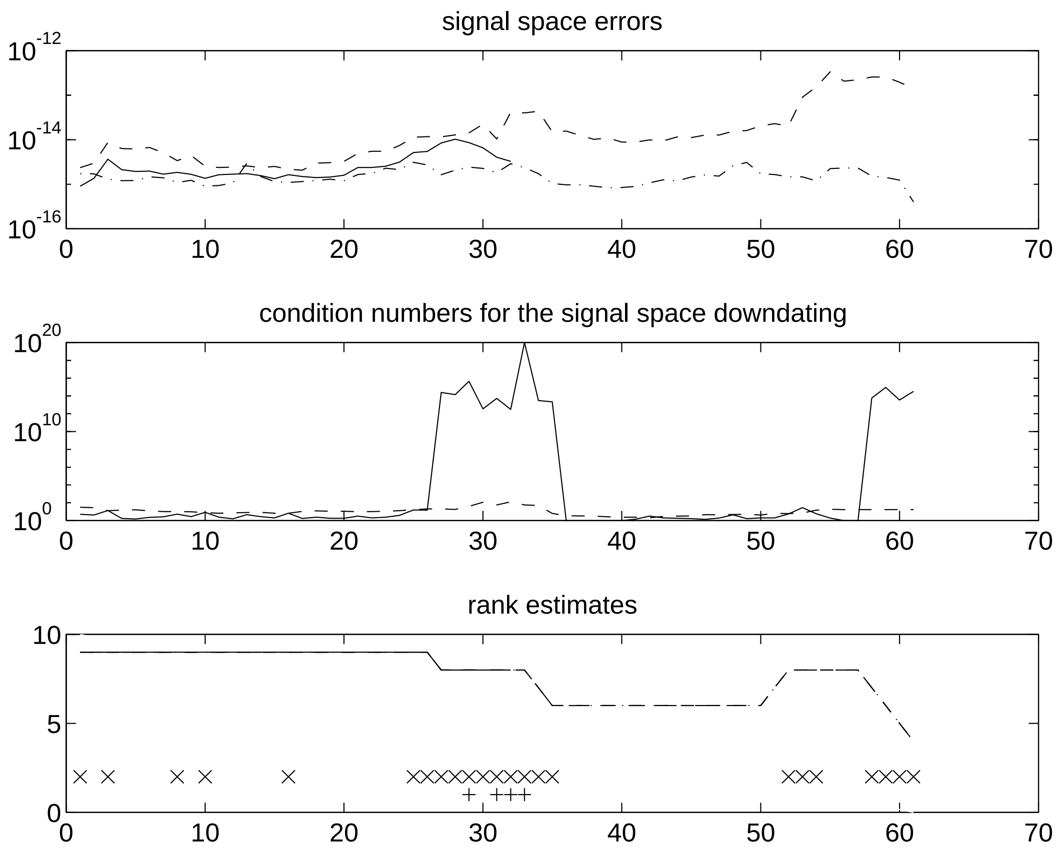

FIG. 5.3. *Test II. Signal space errors, downdating conditioning, and rank estimates for* 13×9 *window matrices.*

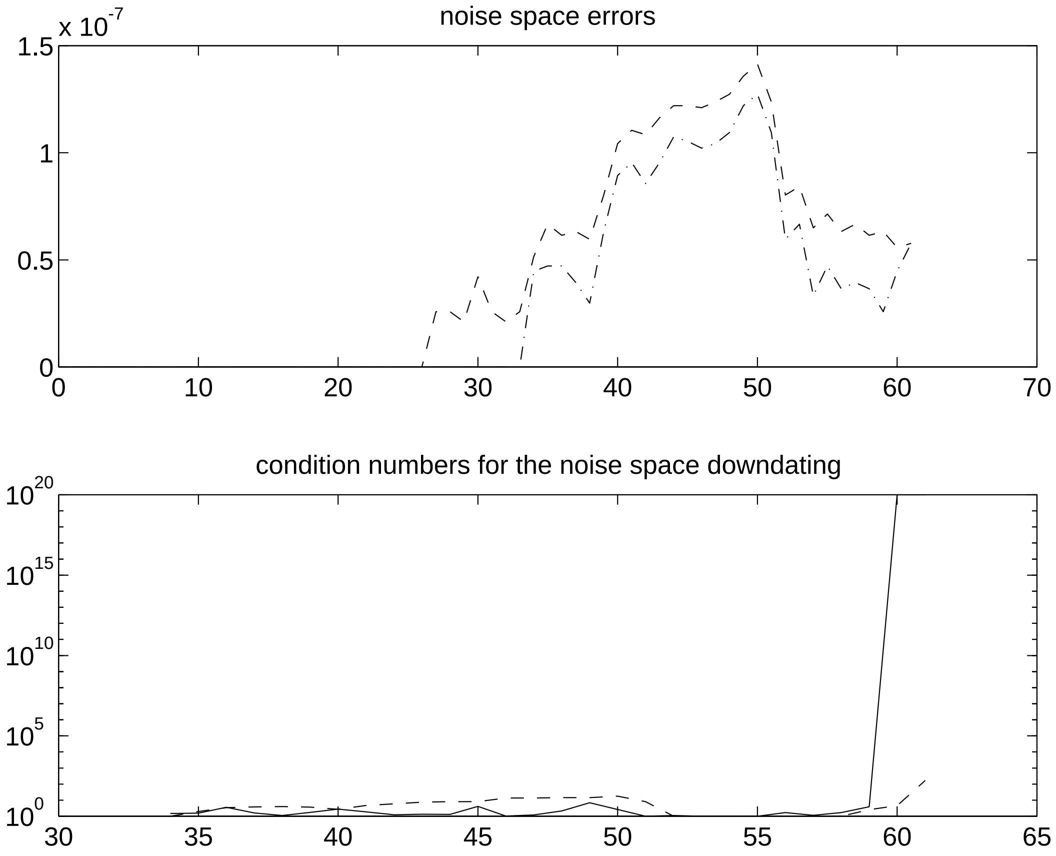

Fig. 5.4. *Test II. Noise space error and downdating conditioning.*