SUBSPACE TRACKING USING A CONSTRAINED HYPERBOLIC URV DECOMPOSITION

Alle-Jan van der Veen

Delft University of Technology, Dept. Electrical Engineering/DIMES, 2628 CD Delft, The Netherlands

The class of Schur subspace estimators provides a parametrization of all minimal-rank matrix approximants that lie within a specified distance of a given matrix, and in particular gives expressions for the column spans of these approximants. In this paper, we derive an updating algorithm for an interesting member of the class, making use of a constrained hyperbolic URV-like decomposition.

1. INTRODUCTION

Fast adaptive subspace estimation plays an increasingly important role in modern signal processing. It forms the key ingredient in many sensor array signal processing algorithms, system identification, and several recently derived blind signal separation and equalization algorithms.

The generic subspace estimation problem in these applications might be stated as follows. Suppose that we are given a data matrix $X : m \times n$, measured column-by-column, that satisfies the model $X = \tilde{X} + \tilde{N}$, where $\tilde{X}$ is a low rank matrix and $\tilde{N}$ is a disturbance. Knowing only X, we can try to estimate $\tilde{X}$ by solving

$$\min_{\hat{X}} \|X - \hat{X}\| \quad \text{s.t. } \operatorname{rank}(\hat{X}) = d \tag{1}$$

where $\|\cdot\|$ denotes the matrix 2-norm (largest singular value). The value of the rank d is either given or is estimated from the singular values of X. The usual truncated SVD (TSVD) solution is to set all but the largest d singular values of X equal to zero. In subspace estimation, we are primarily interested in the column span of $\tilde{X}$. For the TSVD solution, this space is estimated by the span of the first d left singular vectors of X, the so-called principal subspace.

Because directly computing and updating the TSVD is computationally expensive, several other subspace estimators based on cheaper decompositions have been proposed, such as the rank-revealing QR (RRQR, viz. [1]) or the URV [2]. Alternatively, there are efficient subspace tracking and approximate SVD-updating algorithms which under stationary conditions gradually converge towards the principal subspace or the SVD, e.g., [3–5].

A more recent development is the Schur subspace estimation (SSE) technique [6, 7]. It is based on the knowledge of an upper bound to the noise, $\|\tilde{N}\| \le \gamma$, and gives a parametrization for all $\hat{X}$ that satisfy

$$\min_{\hat{X}} \operatorname{rank}(\hat{X}) \quad \text{s.t.} \quad \|X - \hat{X}\| \le \gamma. \tag{2}$$

It is readily shown that the resulting approximants $\hat{X}$ have rank d, where d is equal to the number of singular values of X that are larger than γ. The TSVD is within the class, but it is not explicitly identified. The prime advantage of the SSE technique is that it gives subspace estimates that have the correct dimension and a known performance (in terms of γ), but are substantially easier to compute and update than the TSVD.

The computation of Schur subspace estimates is based on an implicit signed Cholesky factorization

$$XX^* - \gamma^2 I =: BB^* - AA^*$$

where A, B have minimal dimensions. (* denotes the hermitian transpose.) Thus, the spectrum of XX^* is shifted such that the small eigenvalues become negative, which enables their separation from the large eigenvalues. It is readily shown from inertia considerations that, even though A and B are not unique, if X has d singular values larger than γ and $m-d$ less than γ, then B has d columns and A has $m-d$ columns. The main result of [6] is that, *for any such pair* (A, B)*, all principal subspace estimates leading to approximants* $\hat{X}$ *satisfying (2) are given by the column span of* $B - AM$*, where* M *is any matrix of compatible size with* $\|M\| \le 1$. The factorization can be computed via a hyperbolic factorization

$$[\gamma I \quad X]\Theta = [(A \;\; 0) \quad (B \;\; 0)]$$

where Θ is a J-unitary matrix (this notion is defined in section 2).

Straightforward generalizations are possible. Suppose that instead of $\|\tilde{N}\| < \gamma$, we know $\tilde{N}\tilde{N}^* \le \gamma^2 R_N$, where R_N could be an estimate of the noise covariance matrix. An implicit factorization of $XX^* - \gamma^2 R_N$ leads to minimal rank approximants $\hat{X}$ such that $\|R_N^{-1/2}(X - \hat{X})\| \le \gamma$. The subspace estimates are computed from $[N \;\; X]\Theta = [(A \;\; 0) \;\; (B \;\; 0)]$ where N is any matrix such that $NN^* = \gamma^2 R_N$, and are still given by the range of $B - AM$, for any $\|M\| \le 1$. Hence, without extra effort, we can take knowledge of the noise covariance matrix into account. Note that, asymptotically, a suitable N simply consists of scaled sample vectors of the noise process, and can be updated in similar ways as X.

In [6], two subspace estimators within the SSE-class were defined as

$$\text{SSE-1}: \quad U_{SSE1} = B \tag{3}$$

$$\text{SSE-2}: \quad U_{SSE2} = B - AM_\Theta, \quad M_\Theta = [I_{m-d} \;\; 0]\Theta_{11}^{-1}\Theta_{12}\begin{bmatrix} I_d \\ 0 \end{bmatrix} \tag{4}$$

The "central" ($M = 0$) estimator SSE-1 is simple to compute and straightforward to update using hyperbolic rotations, but it is biased in cases with zero noise and $\gamma > 0$. SSE-2 is designed to give $\hat{X} = X$ in cases with zero noise, is almost always more accurate than SSE-1, and simulation results are remarkably close to the TSVD subspace estimate [6, 7]. It is "unbiased" in the sense that $\operatorname{ran}(U_{SSE2}) \subset \operatorname{ran}(X)$, and satisfies $\|U_{SSE2}\| \le \|X\|$: the basis is nicely bounded [6]. However, updating SSE-2 is apparently non-trivial: Θ has growing dimensions and a direct implementation of (4) is prohibitive.

In this paper, we present a new SSE-2 updating algorithm. We first show that SSE-2 subspace estimates are obtained from a certain constrained "hyperbolic URV" decomposition. This decomposition can be updated and downdated efficiently, without storing Θ, and requires approximately $2m^2$ rotations plus m^2 multiplications per update vector of dimension m. (A spherical updating variant can reduce this further to order md.) In addition, the principal subspace estimate takes the form of an *orthonormal* basis.

IEEE ICASSP'98, Seattle, May 1998

Table 1. Elementary J-unitary zeroing rotations

In: $[r\ \ x]$ with signature $\tilde{j}_1$;
Out: $\tilde{\theta}$, $\tilde{j}_2$ such that $[r\ \ x]\tilde{\theta} = [r'\ \ 0]$, $\tilde{\theta}\tilde{j}_2\tilde{\theta}^* = \tilde{j}_1$
1. $\tilde{j}_1 = \begin{bmatrix} 1 & \\ & -1 \end{bmatrix}, \lvert r\rvert > \lvert x\rvert \Rightarrow \ \tilde{j}_2 = \begin{bmatrix} 1 & \\ & -1 \end{bmatrix},\ s = \frac{x}{r},\ \tilde{\theta} = \begin{bmatrix} 1 & -s \\ -s^* & 1 \end{bmatrix} \frac{1}{c}$
2. $\tilde{j}_1 = \begin{bmatrix} 1 & \\ & -1 \end{bmatrix}, \lvert r\rvert < \lvert x\rvert \Rightarrow \ \tilde{j}_2 = \begin{bmatrix} -1 & \\ & 1 \end{bmatrix},\ s = \frac{r}{x},\ \tilde{\theta} = \begin{bmatrix} -s^* & 1 \\ 1 & -s \end{bmatrix} \frac{1}{c}$
3. $\tilde{j}_1 = \begin{bmatrix} -1 & \\ & 1 \end{bmatrix}, \lvert r\rvert < \lvert x\rvert \Rightarrow \ \tilde{j}_2 = \begin{bmatrix} 1 & \\ & -1 \end{bmatrix},\ s = \frac{r}{x},\ \tilde{\theta} = \begin{bmatrix} -s^* & 1 \\ 1 & -s \end{bmatrix} \frac{1}{c}$
4. $\tilde{j}_1 = \begin{bmatrix} -1 & \\ & 1 \end{bmatrix}, \lvert r\rvert > \lvert x\rvert \Rightarrow \ \tilde{j}_2 = \begin{bmatrix} -1 & \\ & 1 \end{bmatrix},\ s = \frac{x}{r},\ \tilde{\theta} = \begin{bmatrix} 1 & -s \\ -s^* & 1 \end{bmatrix} \frac{1}{c}$
5. $\tilde{j}_1 = \begin{bmatrix} 1 & \\ & 1 \end{bmatrix} \quad \Rightarrow \ \tilde{j}_2 = \begin{bmatrix} 1 & \\ & 1 \end{bmatrix},\ s = \frac{x}{\lVert [r\ x] \rVert},\ \tilde{\theta} = \begin{bmatrix} c^* & -s \\ s^* & c \end{bmatrix}$
6. $\tilde{j}_1 = \begin{bmatrix} -1 & \\ & -1 \end{bmatrix} \quad \Rightarrow \ \tilde{j}_2 = \begin{bmatrix} -1 & \\ & -1 \end{bmatrix},\ s = \frac{x}{\lVert [r\ x] \rVert},\ \tilde{\theta} = \begin{bmatrix} c^* & -s \\ s^* & c \end{bmatrix}$
where $c = \sqrt{1 - \lvert s\rvert^2}$

2. J-UNITARY MATRICES

At this point, we review some material on J-unitary matrices from [6]. A square matrix Θ is J-unitary if it satisfies $\Theta^* J \Theta = J$, $\Theta J \Theta^* = J$, where J is a signature matrix which follows some prescribed $(p+q)\times(p+q)$ block-partitioning of Θ:

$$\Theta = \begin{array}{c} \\ p \\ q \end{array} \!\!\begin{array}{c} \begin{array}{cc} p & q \end{array} \\ \begin{bmatrix} \Theta_{11} & \Theta_{12} \\ \Theta_{21} & \Theta_{22} \end{bmatrix} \end{array}, \qquad J = \begin{bmatrix} I_p & \\ & -I_q \end{bmatrix}. \tag{5}$$

If Θ is applied to a block-partitioned matrix $[A\ \ B]$, then $[A\ \ B]\Theta = [C\ \ D] \Rightarrow AA^* - BB^* = CC^* - DD^*$. Hence, J associates a positive signature to the columns of A, C, and a negative signature to those of B, D.

For updating purposes, it is necessary to work with column permutations of $[A\ \ B]$ and $[C\ \ D]$, which induces row and column permutations of Θ. Thus we introduce matrices $\tilde{\Theta}$ that are J-unitary with respect to *unsorted* signature matrices $\tilde{J}$ (the tilde reminds of the absence of sorting), satisfying $\tilde{\Theta}^* \tilde{J}_1 \tilde{\Theta} = \tilde{J}_2$, $\tilde{\Theta}\tilde{J}_2\tilde{\Theta}^* = \tilde{J}_1$, where $\tilde{J}_1$ and $\tilde{J}_2$ are diagonal matrices with diagonal entries equal to ± 1. If $M\tilde{\Theta} = N$, then $M\tilde{J}_1 M^* = N\tilde{J}_2 N^*$, so that $\tilde{J}_1$ associates its signature to the columns of M, and $\tilde{J}_2$ associates its signature to the columns of N. By inertia, the total number of positive entries in $\tilde{J}_1$ has to be equal to that in $\tilde{J}_2$, and likewise for the negative entries.

A 2×2 matrix $\tilde{\theta}$ is an elementary J-unitary rotation if it satisfies $\tilde{\theta}^* \tilde{j}_1 \tilde{\theta} = \tilde{j}_2$, $\tilde{\theta}\tilde{j}_2\tilde{\theta}^* = \tilde{j}_1$, for unsorted signature matrices $\tilde{j}_1$, $\tilde{j}_2$. Similar to Givens rotations, it can be used to zero specific entries of vectors: for a given vector $[r\ \ x]$ and signature $\tilde{j}_1$, we can find $\tilde{\theta}$, r', and $\tilde{j}_2$ such that $[r\ \ x]\tilde{\theta} = [r'\ \ 0]$. The precise form that $\tilde{\theta}$ assumes depends on $\tilde{j}_1$ and whether $\lvert r\rvert > \lvert x\rvert$ or $\lvert r\rvert < \lvert x\rvert$, as listed in table 1. Cases 5 and 6 in the table occur when $\tilde{j}_1$ is definite and lead to ordinary circular (unitary) rotations. Situations where $\lvert r\rvert = \lvert x\rvert$ with an indefinite signature $\tilde{j}_1$ are degenerate ($c = 0$): the result $[0\ \ 0]$ is well defined but θ is unbounded.

3. A HYPERBOLIC URV DECOMPOSITION

Let $N : m\times n_1$ and $X : m\times n_2$ be given matrices. We consider implicit factorizations of $XX^* - NN^*$ as

$$XX^* - NN^* \ = \ BB^* - AA^*, \tag{6}$$

where A and B together have m columns. A and B follow from the factorization

$$m \begin{array}{c} \begin{array}{cc} n_1 & n_2 \\ + & - \end{array} \\ \begin{bmatrix} N & X \end{bmatrix} \end{array} \Theta = m \begin{array}{c} \begin{array}{cc} n_1 & n_2 \\ + & - \end{array} \\ \begin{bmatrix} A' & B' \end{bmatrix} \end{array}; \tag{7}$$

$$A' = m \begin{array}{c} \begin{array}{cc} m-d & n_1-m+d \end{array} \\ \begin{bmatrix} A & 0 \end{bmatrix} \end{array}, \quad B' = m \begin{array}{c} \begin{array}{cc} d & n_2-d \end{array} \\ \begin{bmatrix} B & 0 \end{bmatrix} \end{array}$$

where Θ is a J-unitary matrix partitioned conforming to the equation. The factorization always exists although Θ will be unbounded when $XX^* - NN^*$ is singular [6]. However, the factorization is not unique.

One way to find a factorization (7) is by the hyperbolic QR factorization (HQR) [6,8,9]

$$[\overset{+}{N}\quad \overset{-}{X}]\tilde{\Theta} = [\overset{\pm}{R}\quad \overset{\pm}{0}_{m\times(n1+n2-m)}], \tag{8}$$

in which R is a lower or upper triangular $m\times m$ matrix, and $\tilde{\Theta}$ is $(\tilde{J}_1, \tilde{J}_2)$-unitary. Here, $\tilde{J}_1 = J = I_{n_1} \oplus -I_{n_2}$ is given from the outset, and $\tilde{J}_2$ is a resulting *unsorted* signature matrix (signified by $\pm$ in (8)), which is determined along with R. $\tilde{J}_2$ specifies the signature of the columns of R and hence their membership in either A or B. Although this factorization is simple to update, it has the drawback that it does not always exist [6]: the triangular form of R is too restrictive. The set of exceptions is finite, but in the neighborhood of an exception it may happen that A and B are unbounded with nearly collinear column spans.

To get around this, introduce a QR factorization of $[A\ \ B]$: $R = [R_A\quad R_B] = Q^*[A\quad B]$, where R is triangular and Q is unitary. This leads to the more general two-sided decomposition

$$Q^*[\overset{+}{N}\quad \overset{-}{X}]\Theta = [\overset{+}{R_A}\quad \overset{+}{0}\ \mid\ \overset{-}{R_B}\quad \overset{-}{0}]. \tag{9}$$

Note that still $[A\ \ 0 \mid B\ \ 0] = [N\ \ X]\Theta$. This two-sided decomposition always exists. We can choose to have R upper triangular or lower triangular, or even permute the columns of $[A\ \ B]$ before introducing the QR factorization. It is convenient to take R lower triangular: if we split $Q = [Q_A\ \ Q_B]$ accordingly, then

$$\operatorname{ran}(B) = \operatorname{ran}(Q_B).$$

Hence, for this choice, Q_B is an *orthonormal* basis of the (central) principal subspace estimate. If our objective is to estimate a null space basis, then we would swap (A, B) or take R upper triangular so that $\operatorname{ran}(A) = \operatorname{ran}(Q_A)$.

We are interested in SSE-2 subspace estimates, as defined in (4). This definition involves the inversion of submatrices of Θ. We will now show how this can be avoided by posing additional structural restrictions on Θ, which is possible because A, B and Θ are not unique. We can use this freedom to transform M_Θ in (4) to zero, as shown in the following lemma.

Lemma 1. *For given A, B, Θ, consider a transformation by a J-unitary matrix Θ_M:*

$$[A\ \ 0 \mid B\ \ 0]\Theta_M \ = \ [A'\ \ 0 \mid B'\ \ 0] \tag{10}$$
$$\Theta\Theta_M \ = \ \Theta' \tag{11}$$

where Θ_M only acts on the columns of A, B (and corresponding columns of Θ).

Then $\operatorname{ran}(B - AM_\Theta) = \operatorname{ran}(B' - A'M_{\Theta'})$, *i.e., the SSE-2 subspace is invariant under Θ_M. Furthermore, there exists a Θ_M such that $M_{\Theta'} = 0$, i.e., such that* $\operatorname{ran}(B')$ *is the SSE-2 subspace.*

The proof is rather technical and omitted.

Hence, there is a matrix Θ_M which transforms Θ to $\Theta' = \Theta\Theta_M$, such that after the transformation we simply take B' and have the desired SSE-2 subspace basis. Knowing this, there are easier ways to find this transformation. Suppose $[\Theta_{11} \;\; \Theta_{12}]$ is partitioned as

$$[\Theta_{11} \quad \Theta_{12}] = \begin{array}{c} \\ {\scriptstyle m-d} \\ {\scriptstyle n_1-(m-d)} \end{array} \begin{array}{c} \begin{array}{cccc} {\scriptstyle m-d} & {\scriptstyle n_1-(m-d)} & {\scriptstyle d} & {\scriptstyle n_2-d} \end{array} \\ \left[\begin{array}{cc|cc} (\Theta_{11})_{11} & (\Theta_{11})_{12} & (\Theta_{12})_{11} & (\Theta_{12})_{12} \\ * & * & * & * \end{array}\right] \end{array}$$

From the definition of M_Θ in (4), it is seen that to have $M_{\Theta'} = 0$, it suffices to find a transformation on Θ such that $\Theta_{11}'^{-1}\Theta_{12}'$ has a zero (11)-block. This will be the case, for example, if both $(\Theta_{11}')_{12} = 0$ and $(\Theta_{12}')_{11} = 0$. The latter can always be effected by a suitably chosen Θ_M which cancels $(\Theta_{12})_{11}$ against $(\Theta_{11})_{11}$. However, to apply lemma 1, Θ_M is not allowed to change the columns of $(\Theta_{11})_{12}$. To zero this block, we may apply any invertible transformation T_e to the *rows* of $[\Theta_{11} \;\; \Theta_{12}]$:

$$[\Theta_{11}' \quad \Theta_{12}'] = T_e[\Theta_{11} \quad \Theta_{12}]$$

because $\Theta_{11}'^{-1}\Theta_{12}' = \Theta_{11}^{-1}\Theta_{12}$ is invariant under T_e. This leads to a new characterization of SSE-2 estimates:

Theorem 2. *The following factorization always exists and provides an SSE-2 subspace estimate. For given $N : m \times n_1$, $X : m \times n_2$, with $n_1 \ge m$, find the subspace dimension d, Q (unitary), Θ (J-unitary), $R = [R_A \; R_B]$ (lower triangular), $T : (m-d) \times n_1$ (full rank) such that*

$$Q^* \begin{array}{c} \begin{array}{cc} {\scriptstyle n_1} & {\scriptstyle n_2} \\ + & - \end{array} \\ \left[\begin{array}{cc} N & X \end{array}\right] \end{array} \Theta = \begin{array}{c} \begin{array}{cccc} {\scriptstyle m-d} & {\scriptstyle n_1-(m-d)} & {\scriptstyle d} & {\scriptstyle n_2-d} \\ + & + & - & - \end{array} \\ \left[\begin{array}{cccc} R_A & 0 & R_B & 0 \end{array}\right] \end{array}, \tag{12}$$

$$T \begin{array}{c} \begin{array}{cc} {\scriptstyle n_1} & {\scriptstyle n_2} \\ + & - \end{array} \\ \left[\begin{array}{cc} I & 0 \end{array}\right] \end{array} \Theta = \begin{array}{c} \begin{array}{cccc} {\scriptstyle m-d} & {\scriptstyle n_1-(m-d)} & {\scriptstyle d} & {\scriptstyle n_2-d} \\ + & + & - & - \end{array} \\ \left[\begin{array}{cccc} I & 0 & 0 & * \end{array}\right] \end{array}. \tag{13}$$

With the partitioning $Q = [Q_A \;\; Q_B]$, an orthonormal basis for the SSE-2 subspace estimate is given by Q_B.

By virtue of [6, thm. 2.1], the above factorization always exists. If $XX^* - NN^*$ is singular, then certain columns of Θ are unbounded and corresponding columns of R are identically zero.

The factorization in (12) is reminiscent of the URV decomposition [2], but with a J-unitary Θ. The following corollary shows that the constraint (13) ensures certain desirable norm properties.

Corollary 3. *The factorization (12)-(13) is such that* $\operatorname{ran}(Q_B) \subset \operatorname{ran}(X)$, $\|R_B\| \le \|X\|$, $\|R_A\| \le \|N\|$.

PROOF Using the fact that $M_\Theta = 0$, lemma 3.4 in [6] implies $BB^* \le XX^*$, $AA^* \le NN^*$. It remains to apply the definition $[A \;\; B] = [Q_A \;\; Q_B][R_A \;\; R_B]$ where Q is unitary and R is lower. □

4. UPDATING THE SSE-2

Now that we have identified (12)-(13) as a factorization which provides an SSE-2 subspace, we investigate how this factorization can be updated when new columns for X and N become available. The update consists of two phases, one to update (12), and a second to restore the zero structure of (13). Several updating algorithms are possible, depending on one's objectives.

4.1. Updating $Q^*[N \;\; X]\Theta$

Suppose we have already computed the decomposition $Q^*[N \;\; X]\tilde{\Theta} = [R \;\; 0]$, where $R = [R_A \;\; R_B]$ is lower triangular and sorted according to signature. In principle, updating the factorization with a new column $\mathbf{x}$ or $\mathbf{n}$ is straightforward. Indeed, let us say that we want to find a new factorization $Q'^*[N' \;\; X']\tilde{\Theta}' = [R' \;\; 0]$, where either $N' = [N \;\; \mathbf{n}]$ or $X' = [X \;\; \mathbf{x}]$. Making use of the previously computed decomposition, it suffices to find Q_c and $\tilde{\Theta}_c$ such that

$$Q_c^* \begin{array}{c} \begin{array}{ccc} {\scriptstyle m-d} & {\scriptstyle d} & {\scriptstyle 1} \\ + & - & j_c \end{array} \\ \left[\begin{array}{ccc} R_A & R_B & \mathbf{c} \end{array}\right] \end{array} \tilde{\Theta}_c = \begin{array}{c} \begin{array}{ccc} {\scriptstyle m-d'} & {\scriptstyle d'} & {\scriptstyle 1} \\ + & - & j_c' \end{array} \\ \left[\begin{array}{ccc} R_A' & R_B' & 0 \end{array}\right] \end{array} \tag{14}$$
$$Q' := QQ_c\,,$$

where $\mathbf{c} = Q^*\mathbf{n}$ or $\mathbf{c} = Q^*\mathbf{x}$. (Note that we need to store and update Q to apply this transformation. Storage of $\tilde{\Theta}$ will not be needed.) In the first case, $\mathbf{c}$ has a positive signature $j_c = 1$; in the second case, $j_c = -1$. Denote the signature of R by $J = I_{m-d} \oplus -I_d$.

Table 2. Two ways to zero c_i

col:	1. Compute $\tilde{\theta}$ and $\tilde{j}_2$ s.t. $[R_{i,i} \;\; c_i]\tilde{\theta} = [* \;\; 0]$ 2. Apply $\tilde{\theta}$ to the i-th column of R and $\mathbf{c}$; update signatures
row:	1. Determine q s.t. $q^*\left[\begin{smallmatrix} c_i \\ c_{i+1} \end{smallmatrix}\right] = \left[\begin{smallmatrix} 0 \\ * \end{smallmatrix}\right]$. 2. Apply q^* to rows $(i, i+1)$ of R; apply q to Q 3. Compute $\tilde{\theta}$ and $\tilde{j}_2$ s.t. $[R_{i,i} \;\; R_{i,i+1}]\tilde{\theta} = [* \;\; 0]$ 4. Apply $\tilde{\theta}$ to columns $(i, i+1)$ of R; update signatures

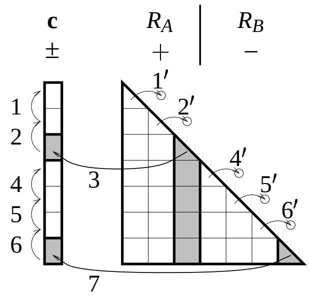

Figure 1. Order in which zero entries are created by algorithm *zero-c*. Only column operations (rotations 3 and 7) are possibly hyperbolic and may lead to signature changes

To compute the factorization (14), the entries $c_1, c_2, \cdots, c_m$ of $\mathbf{c}$ are zeroed in turn. As listed in table 2, there are two possibilities to do this: by elementary column rotations $\tilde{\theta}$ or by elementary row rotations q. The "*col*" scheme to zero entry c_i is the most natural and efficient, and directly zeros c_i against $R_{i,i}$. The "*row*" scheme first computes an elementary circular (unitary) rotation q to zero c_i against c_{i+1}, and then a $\tilde{\theta}$-rotation to zero the resulting fill-in in $R_{i,i+1}$ against $R_{i,i}$. For reasons of numerical stability, it is desirable to minimize the number of *hyperbolic* rotations, i.e. rotations $\tilde{\theta}$ that act on columns with unequal signatures. Hence, we propose to zero most entries c_i using row operations, in spite of the added complexity, and to use column operations only for zeroing c_{m-d} and c_m.

A graphical representation of this scheme is given in figure 1. Hyperbolic rotations and signature changes are only possible in steps $m-d$ and m. The θ-rotations in the row stages act on columns of equal signatures, so that they are circular rotations without signature changes. The resulting signature of R depends on the initial and final signature of $\mathbf{c}$, i.e., j_c and j_c': see figure 2.

The second phase is to restore the sorting of the columns of R according to their signature. This is only necessary in cases (b) and (d_2) of figure 2, and it suffices to move the last column of R by a series of d swaps with its right neighbors. After each permutation, the resulting fill-in in $R_{i,i+1}$ has to be zeroed by a q-rotation. If desired, this phase can be made data-independent by always performing the permutations, independent of the signatures.

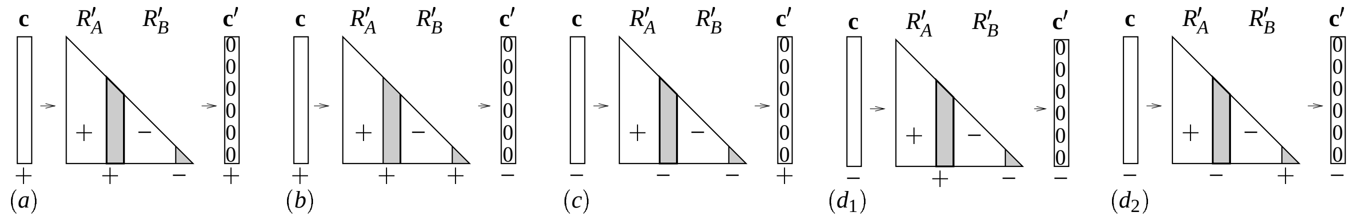

Figure 2. The four possible signature changes of **c**, **c′**, and the resulting possible signatures J' (after *zero-c*, before sorting). Only columns $m-d$ and d of R may have changed signature.

Table 3. SSE-2 updating algorithm

In: **c**, j_c; R (lower), $J = \text{diag}[J_1, \cdots, J_m]$ (sorted), Q (unitary); d
Out: updated versions of R, J, Q, d

Algorithm SSE2-update:

zero-c: $\mathbf{c} := Q^*\mathbf{c}$
  $e_c = 0,\ e_{m-d} = 1,\ e_m = 0,$
  for $i = 1$ to m
    if $i = m-d$ or $i = m$
      zero c_i using **col** $(\tilde{\theta})$
      $[e_i\ \ e_c] := [e_i\ \ e_c]\tilde{\theta}$
    else
      zero c_i using **row**
  end

sort-R: for $i = m-1$ down to $m-d+1$
    permute columns i and $i+1$ of R (and J_i, J_{i+1})
    compute q to zero the fill-in $R_{i,i+1}$ against $R_{i+1,i+1}$
    apply q to rows $(i, i+1)$ of R and columns of Q
  end

M_Θ-trans.: if $J_{m-d} = -1$ and $J_{m-d+1} = +1$, (case (d))
    compute $\tilde{\theta}, \tilde{j}_2$ s.t. $[e_{m-d}\ \ e_m]\tilde{\theta} = [*\ \ 0]$
    apply $\tilde{\theta}$ to columns $(m-d, m-d+1)$ of R
    compute q to zero fill-in $R_{m-d,m-d+1}$
    apply q to rows $(m-d, m-d+1)$ of R and to Q
  end
  update d : $d := d + \frac{1}{2}(j'_c - j_c)$

(A matlab implementation is available upon request.)

4.2. Restoring the structure of Θ

The next step is to modify the candidate Q_c and $\tilde{\Theta}_c$ by some Q_M and $\tilde{\Theta}_M$ in order to satisfy the structural conditions (13) on Θ. Equation (13) shows that we do not have to keep track of T and Θ at all: we only have to update a matrix $[I_{m-d}\ \ 0_{m-d\times d}]$. The columns marked '$*$' in (13) never change, so we do not have to track them. Obviously, we do not have to store $[I_{m-d}\ \ 0]$. Hence, updating is possible by only storing matrices Q and $R = [R_A\ \ R_B]$.

With regard to the zero-structure, there are four possibilities, which match the four cases (a)–(d) we had before. The investigation of each of the cases separately is technical and omitted, but the result is simple: only in case (d), i.e., $j_c = -1$, $j'_c = -1$, an action involving $\tilde{\Theta}_M$ is required. Moreover, it suffices to keep track of only three entries of $[I\ \ 0]$ during the update, which will be denoted by e_{m-d}, e_m, e_c.

The resulting algorithm is summarized in table 3. The M_Θ-transformation, if needed, consists of a single θ-rotation on the columns of R, followed by a q-rotation on the rows of R to zero the fill-in. The sorting stage sorts unconditionally for simplicity and uniformity, and only up to column $m-d+1$.

The updating algorithm can be initialized by $R = 0$, $d = 0$, $Q = I_m$. The computational complexity is assessed as m^2 multiplications (for the initial transformation of **c** by Q), and about $2m^2 + 2md$ elementary rotations. This is four times more than the original HQR scheme for computing the SSE-1.

5. DISCUSSION

The updating algorithm which we derived has the following properties. Its main feature is a localized, piecewise regular, data-independent computational flow using plane J-unitary rotations. The algorithm consists of two phases: a forward phase to zero the update vector, and a backward phase to restore the sorting and at the same time satisfy a structural constraint. Each phase is fully pipelineable, but unfortunately the combination is not, unless they can be meshed together (with some effort, this is sometimes possible, cf. [4]). Per update vector, there are at most 3 *hyperbolic* rotations, which is not minimal, but significantly less than the HQR updating algorithm proposed previously in [6]. Updating and downdating uses the same computational structure, since downdating X by a vector **x** can be done by updating N by **x**. Exponential windowing and several interesting updating/downdating schemes are possible, which will be reported separately.

References

[1] S. Chandrasekaran and I.C.F. Ipsen, "On rank-revealing factorisations," *SIAM J. Matrix Anal. Appl.*, vol. 15, pp. 592–622, Apr. 1994.

[2] G.W. Stewart, "An updating algorithm for subspace tracking," *IEEE Tr. Signal Proc.*, vol. 40, pp. 1535–1541, June 1992.

[3] P. Comon and G.H. Golub, "Tracking a few extreme singular values and vectors in signal procesing," *Proceedings of the IEEE*, vol. 78, pp. 1327–1343, Aug. 1990.

[4] M. Moonen, P. Van Dooren, and F. Vanpoucke, "On the QR algorithm and updating the SVD and URV decomposition in parallel," *Lin. Alg. Appl.*, vol. 188/189, pp. 549–568, July 1993.

[5] B. Yang, "Projection approximation subspace tracking," *IEEE Trans. Signal Processing*, vol. 43, pp. 95–107, Jan. 1995.

[6] A.J. van der Veen, "A Schur method for low-rank matrix approximation," *SIAM J. Matrix Anal. Appl.*, vol. 17, pp. 139–160, Jan. 1996.

[7] J. Götze and A.J. van der Veen, "On-line subspace estimation using a Schur-type method," *IEEE Trans. Signal Processing*, vol. 44, pp. 1585–1589, June 1996.

[8] A. Bunse-Gerstner, "An analysis of the HR algorithm for computing the eigenvalues of a matrix," *Lin. Alg. Appl.*, vol. 35, pp. 155–173, 1981.

[9] R. Onn, A.O. Steinhardt, and A.W. Bojanczyk, "The hyperbolic singular value decomposition and applications," *IEEE Trans. Signal Processing*, vol. 39, pp. 1575–1588, July 1991.