Subspace-Based Noise Reduction for Speech Signals via Diagonal and Triangular Matrix Decompositions*

Per Christian Hansen[a] and **Søren Holdt Jensen**[b]

[a]Informatics and Mathematical Modelling, Building 321
Technical University of Denmark, DK-2800 Lyngby, Denmark

[b]Multimedia Information and Signal Processing, Department of Electronic Systems,
Aalborg University, Niels Jernes Vej 12, DK-9220 Aalborg, Denmark

We survey the definitions and use of rank-revealing matrix decompositions in single-channel noise reduction algorithms for speech signals. Our algorithms are based on the rank-reduction paradigm and, in particular, signal subspace techniques. The focus is on practical working algorithms, using both diagonal (eigenvalue and singular value) decompositions and rank-revealing triangular decompositions (ULV, URV, VSV, ULLV and ULLIV). In addition we show how the subspace-based algorithms can be evaluated and compared by means of simple FIR filter interpretations. The algorithms are illustrated with working Matlab code and applications in speech processing.

Key words: rank reduction, subspace methods, noise reduction, speech processing, SVD, GSVD, rank-revealing decompositions, FIR filter interpretation, canonical filters.

1. INTRODUCTION

The signal subspace approach has proved itself useful for signal enhancement in speech processing and many other applications – see, e.g., the recent survey [10]. The area has grown dramatically over the last 20 years, along with advances in efficient computational algorithms for matrix computations [16], [46], [47], especially singular value decompositions and rank-revealing decompositions.

The central idea is to approximate a matrix, derived from the noisy data, with another matrix of *lower rank* from which the reconstructed signal is derived. As stated in [42]: *"Rank reduction is a general principle for finding the right trade-off between model bias and model variance when reconstructing signals from noisy data."*

Throughout the literature of signal processing and applied mathematics these methods are formulated in terms of different notations, such as eigenvalue decompositions, Karhunen-Loève transformations, and singular value decompositions. All these formulations are mathematically equivalent, but nevertheless the differences in notation can be an obstacle to understanding and using the different methods in practise.

Our goal is to survey the underlying mathematics and present the techniques and algorithms in a common framework and a common notation. In addition to methods based on diagonal (eigenvalue and singular value) decompositions, we survey the use of rank-revealing triangular decompositions. Within this framework we also discuss alternatives to the classical least-squares formulation, and we show how signals with general (non-white) noise are treated by explicit and, in particular, implicit prewhitening. Throughout the paper we provide small working Matlab codes that illustrate the algorithms and their practical use.

We focus on *signal enhancement* methods which directly estimate a clean signal from a noisy one (we do not estimate parameters in a parameterized signal model). Our presentation starts with formulations based on (estimated) covariance matrices, and makes extensive use of eigenvalue decompositions as well as the ordinary and generalized singular value decomposition (SVD and GSVD) – the latter also referred to as the quotient SVD (QSVD). All these subspace techniques originate from the seminal 1982 paper [49] by Tufts and Kumaresan, who considered noise reduction of signals consisting of sums of damped sinusoids via linear prediction methods.

Early theoretical and methodological developments in SVD-based least-squares subspace methods for signals with white noise were given in the late 80s and early 90s by Cadzow [6], De Moor [7], Scharf [41], and Scharf and Tufts [42]. Dendrinos, Bakamidis and Carayannis [9] used these techniques for speech signals, and Van Huffel [52] applied a similar approach – using the minimum variance estimates from [7] – to exponential data modelling. Other applications of these methods can be found, e.g., in [10], [12], [27] and [40]. Techniques for general noise, based on the GSVD, originally appeared in [30], and some applications of these methods can be

*This work was supported in part by the Danish Research Council under Grants 21-03-0574 and 26-02-0092.

found in [25], [26], [32] and [38].

Next we describe computationally favorable alternatives to the (G)SVD methods, based on rank-revealing triangular decompositions. The advantages of these methods are faster computation and faster up- and downdating, which is important in dynamic signal processing applications. This class of algorithms originates from work by Moonen, Van Dooren and Vandewalle [39] on approximate SVD updating algorithms, and in particular Stewart's work on URV and ULV decompositions [44], [45]. Some applications of these methods can be found in [1], [33] (direction-of-arrival estimation) and [53] (total least squares). We also describe some extensions of these techniques to rank-revealing ULLV decompositions of pairs of matrices, originating in works by Luk and Qiao [34], [36] and Bojanczyk [4].

Further extensions of the GSVD and ULLV algorithms to rank-deficient noise, typically arising in connection with narrow-band noise and interference, were described in recent work by Zhong, Li and Tai [55] and ourselves [20], [22].

Finally we show how all the above algorithms can be interpreted in terms of FIR filters defined from the decompositions involved [11], [21], and we introduce a new analysis tool called "canonical filters" which allows us to compare the behavior and performance of the subspace-based algorithms in the frequency domain. The hope is that this theory can help to bridge the gap between the matrix notation and more classical signal processing terminology.

Throughout the paper we make use of the important concept of *numerical rank* of a matrix. The numerical rank of a matrix H with respect to a given threshold τ is the number of columns of H that are guaranteed to be linearly independent for any perturbation of H with norm less than τ. In practise, the numerical rank is computed as the number of singular values of H greater than τ. We refer to [19], [43] and [31] for motivations and further insight about this issue.

We stress that we do not try to cover all aspects of subspace methods for signal enhancement. For example, we do not treat a number of heuristic methods such as the spectral domain constrained estimator [12], as well as extensions that incorporate various perceptual constraints [28], [54].

A few words about the notation used throughout the paper: $\mathcal{E}(\cdot)$ denotes expectation; $\mathcal{R}(A)$ denotes the range (or column space) of the matrix A; $\sigma_i(A)$ denotes the ith singular value of A; A^T denotes the transpose of A, and $A^{-T} = (A^{-1})^T = (A^T)^{-1}$, I_q is the identity matrix of order q; and $\mathcal{H}(v)$ is the Hankel matrix with n columns defined from the vector v (see (4)).

2. THE SIGNAL MODEL

Throughout this paper we consider only wide-sense stationary signals with zero mean, and a *digital signal* is always a column vector $s \in \mathbb{R}^n$ with $\mathcal{E}(s) = 0$. Associated with s is an $n \times n$ symmetric positive semidefinite *covariance matrix*, given by $C_s \equiv \mathcal{E}(s\, s^T)$; this matrix has Toeplitz structure, but we do not make use of this property. We shall make some important assumptions about the signal.

The Noise Model. We assume that the signal s consists of a *pure signal* $\bar{s} \in \mathbb{R}^n$ corrupted by *additive noise* $e \in \mathbb{R}^n$,

$$s = \bar{s} + e, \qquad (1)$$

and that the noise level is not too high, i.e., $\|e\|_2$ is somewhat smaller than $\|\bar{s}\|_2$. In most of the paper we also assume that the covariance matrix C_e for the noise has full rank. Moreover, we assume that we are able to sample the noise, e.g., in periods where the pure signal vanishes (for example, in speech pauses). We emphasize that the sampled noise vector e is not the exact noise vector in (1), but a vector that is statistically *representative* of the noise.

The Pure Signal Model. We assume that the pure signal $\bar{s}$ and the noise e are *uncorrelated*, i.e., $\mathcal{E}(\bar{s}\, e^T) = 0$, and consequently we have

$$C_s = C_{\bar{s}} + C_e. \qquad (2)$$

In the common case where C_e has full rank, it follows that C_s also has full rank (the case $\mathrm{rank}(C_e) < n$ is treated in Section 7). We also assume that the pure signal $\bar{s}$ lies in a proper *subspace* of $\mathbb{R}^n$; i.e.,

$$\bar{s} \in \overline{\mathcal{S}} \subset \mathbb{R}^n, \quad \mathrm{rank}(C_{\bar{s}}) = \dim(\overline{\mathcal{S}}) = k < n. \qquad (3)$$

The central point in subspace methods is this assumption about the pure signal $\bar{s}$ lying in a (low-dimensional) subspace of $\mathbb{R}^n$ called the *signal subspace*. The main goal of all subspace methods is to estimate this subspace and to find a good estimate $\hat{s}$ (of the pure signal $\bar{s}$) in this subspace.

The subspace assumption (which is equivalent to the assumption that $C_{\bar{s}}$ is rank deficient) is satisfied, e.g., when the signal is a sum of (exponentially damped) sinusoids. This assumption is perhaps rarely satisfied exactly for a real signal, but it is a *good model* for many signals, such as those arising in speech processing [37].[2]

For practical computations with algorithms based on the above $n \times n$ covariance matrices, we need to be able to compute estimates of these matrices. The standard way to do this is to assume that we have access to *data*

[2] It is also a good model for NMR signals [50], [51], but these signals are not treated in this paper.

vectors which are longer than the signals we want to consider. For example, for the noisy signal we assume that we know a data vector $s' \in \mathbb{R}^N$ with $N > n$, which allows us to estimate the covariance matrix for s as follows. We note that the length N is often determined by the application (or the hardware in which the algorithm is used).

Let $\mathcal{H}(s')$ be the $m \times n$ Hankel matrix defined from the vector s' as

$$\mathcal{H}(s') = \begin{pmatrix} s'_1 & s'_2 & s'_3 & \cdots & s'_n \\ s'_2 & s'_3 & s'_4 & \cdots & s'_{n+1} \\ s'_3 & s'_4 & s'_5 & \cdots & s'_{n+2} \\ \vdots & \vdots & \vdots & & \vdots \\ s'_m & s'_{m+1} & s'_{m+2} & \cdots & s'_N \end{pmatrix} \tag{4}$$

with $m + n - 1 = N$ and $m \geq n$. Then we define the *data matrix* $H = \mathcal{H}(s')$, such that we can estimate[3] the covariance matrix C_s by

$$C_s \approx \tfrac{1}{m} H^T H. \tag{5}$$

Moreover, due to the assumption about additive noise we have $s' = \bar{s}' + e'$ with $\bar{s}', e' \in \mathbb{R}^N$, and thus we can write

$$H = \overline{H} + E \quad \text{with} \quad \overline{H} = \mathcal{H}(\bar{s}'), \quad E = \mathcal{H}(e'). \tag{6}$$

Similar to the assumption about $C_{\bar{s}}$, we assume that $\mathrm{rank}(\overline{H}) = k$.

In broad terms, the goal of our algorithms is to compute an estimate $\hat{s}$ of the pure signal $\bar{s}$ from measurements of the noisy data vector s' and a representative noise vector e'. This is done via a rank-k estimate $\widehat{H}$ of the Hankel matrix $\overline{H}$ for the pure signal, and we note that we do not require the estimate $\widehat{H}$ to have Hankel structure.

There are several approaches to extracting a signal vector from the $m \times n$ matrix $\widehat{H}$. One approach, which produces a length-N vector $\hat{s}'$, is to *average* along the anti-diagonals of $\widehat{H}$, which we write as

$$\hat{s}' = \mathcal{A}(\widehat{H}) \in \mathbb{R}^N. \tag{7}$$

The corresponding Matlab code is

```
shat = zeros(N,1);
for i=1:N
  shat(i) = mean(diag(fliplr(Hhat),n-i));
end
```

This approach leads to the FIR filter interpretation in §9. The rank-reduction + averaging process can be iterated, and Cadzow [6] showed that this process converges to a rank-k Hankel matrix; however, De Moor [8] showed that this may not be the desired matrix. In practise the single averaging in (7) works well.

[3]Alternatively we could work with the Toeplitz matrices obtained by reversing the order of the columns of the Hankel matrices; all our relations will still hold.

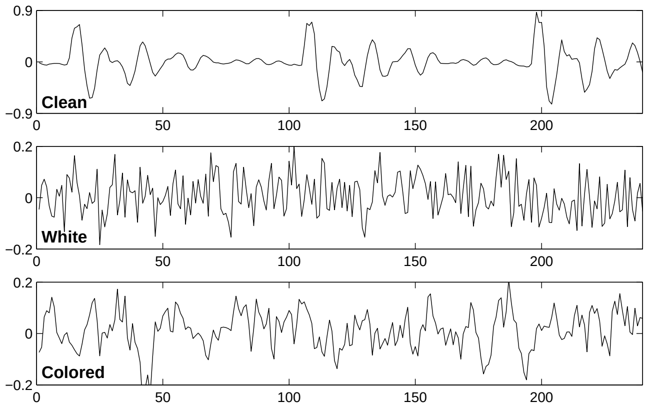

Figure 1. The three signals of length $N = 240$ used in our examples. Top: clean speech signal (voiced segment of male speaker). Middle: white noise generated by Matlab's `randn` function. Bottom: colored noise (segment of a recording of strong wind). The clean signal slightly violates the subspace assumption (3); see Fig. 3.

Doclo and Moonen [10] found that the averaging operation is often unnecessary. An alternative approach, which produces a length-n vector, is therefore to simply *extract* (and transpose) an arbitrary row of the matrix, i.e.,

$$\hat{s} = \widehat{H}(\ell, :)^T \in \mathbb{R}^n, \qquad \ell \text{ arbitrary}. \tag{8}$$

This approach lacks a solid theoretical justification, but due to its simplicity it lends itself well to the up- and downdating techniques in dynamical processing; see §8.

Throughout the paper we illustrate the use of the subspace algorithms with an example from speech processing, where the clean signalis a 30 ms segment of a voiced sound from a male speaker recorded at 8 kHz sampling frequency of length is $N = 240$. We use two noise signals, a white noise signal generated by Matlab's `randn` function, and a segment of a recording of strong wind. All three signals, shown in Fig. 1, can be considered quasi-stationary in the considered segment. We always use $m = 211$ and $n = 30$, and the signal-to-noise ratio in the noisy signals, defined as

$$\mathrm{SNR} = 20 \log\left(\|\bar{s}\|_2 / \|e\|_2\right) \text{ dB},$$

is always 10 dB.

When displaying the spectrum of a signal, we always use the LPC power spectrum computed with Matlab's `lpc` function with order 12, which is standard in speech analysis of signals sampled at 8 kHz.

3. WHITE NOISE: SVD METHODS

To introduce ideas, we consider first the ideal case of white noise, i.e., the noise covariance matrix is a scaled identity,

$$C_e = \eta^2 I_n, \tag{9}$$

where η^2 is the variance of the noise. The covariance matrix for the pure signal has the eigenvalue decomposition

$$C_{\bar{s}} = \overline{V}\,\bar{\Lambda}\,\overline{V}^T, \qquad \bar{\Lambda} = \mathrm{diag}(\bar{\lambda}_1, \ldots, \bar{\lambda}_n) \tag{10}$$

with $\bar{\lambda}_{k+1} = \cdots = \bar{\lambda}_n = 0$. The covariance matrix for the noisy signal, $C_s = C_{\bar{s}} + \eta^2 I_n$, has the same eigenvectors while its eigenvalues are $\bar{\lambda}_i + \eta^2$ (i.e., they are "shifted" by η^2). It follows immediately that given η and the eigenvalue decomposition of C_s we can perfectly reconstruct $C_{\bar{s}}$ simply by subtracting η^2 from the largest k eigenvalues of C_s and inserting these in (10).

In practise, we cannot design a robust algorithm on this simple relationship. For one thing, the rank k is rarely known in advance, and white noise is a mathematical abstraction. Moreover, even if the noise e is close to being white, a practical algorithm must use an estimate of the variance η^2, and there is a danger that we obtain some negative eigenvalues when subtracting the variance estimate from the eigenvalues of C_s.

A more robust algorithm is obtained by replacing k with an *underestimate* of the rank, and by avoiding the subtraction of η^2. The latter is justified by a reasonable assumption that the largest k eigenvalues $\bar{\lambda}_i$, $i = 1, \ldots, k$ are somewhat greater than η^2.

A working algorithm is now obtained by replacing the covariance matrices with their computable estimates. For both pedagogical and computational/algorithmic reasons, it is most convenient to describe the algorithm in terms of the two SVDs

$$\begin{aligned} \overline{H} &= \overline{U}\,\overline{\Sigma}\,\overline{V}^T \\ &= (\,\overline{U}_1\,,\,\overline{U}_2\,)\begin{pmatrix}\overline{\Sigma}_1 & 0\\ 0 & 0\end{pmatrix}(\,\overline{V}_1\,,\,\overline{V}_2\,)^T \end{aligned} \tag{11}$$

and

$$\begin{aligned} H &= U\,\Sigma\,V^T \\ &= (\,U_1\,,\,U_2\,)\begin{pmatrix}\Sigma_1 & 0\\ 0 & \Sigma_2\end{pmatrix}(\,V_1\,,\,V_2\,)^T \end{aligned} \tag{12}$$

in which $\overline{U}, U \in \mathbb{R}^{m\times n}$ and $\overline{V}, V \in \mathbb{R}^{n\times n}$ have orthonormal columns, and $\overline{\Sigma}, \Sigma \in \mathbb{R}^{n\times n}$ are diagonal. These matrices are partitioned such that $\overline{U}_1, U_1 \in \mathbb{R}^{m\times k}$, $\overline{V}_1, V_1 \in \mathbb{R}^{n\times k}$ and $\overline{\Sigma}_1, \Sigma_1 \in \mathbb{R}^{k\times k}$. We note that the SVDs immediately provide the eigenvalue decompositions of the cross-product matrices, because

$$\overline{H}^T\overline{H} = \overline{V}\,\overline{\Sigma}^2\,\overline{V}^T, \qquad H^T H = V\,\Sigma^2\,V^T.$$

The pure signal subspace is then given by $\overline{\mathcal{S}} = \mathcal{R}(\overline{V}_1)$, and our goal is to estimate this subspace and to estimate the pure signal via a rank-k estimate $\widehat{H}$ of the pure-signal matrix $\overline{H}$.

Moving from the covariance matrices to the use of the cross-product matrices, we must make further assumptions [7], namely (in the white-noise case) that the matrices E and $\overline{H}$ satisfy

$$\tfrac{1}{m} E^T E = \eta^2 I_n \qquad \text{and} \qquad \overline{H}^T E = 0. \tag{13}$$

These assumptions are stronger than $C_e = \eta^2 I_n$ and $\mathcal{E}(\bar{s}\,e^T) = 0$. The first assumption is equivalent to the requirement that the columns of $(\sqrt{m}\eta)^{-1}E$ are orthonormal. The second assumption implies the requirement $m \geq n + k$.

Then it follows that

$$\tfrac{1}{m} H^T H = \tfrac{1}{m}\overline{H}^T\overline{H} + \eta^2 I_n \tag{14}$$

and if we insert the SVDs and multiply with m we obtain the relation

$$(V_1\,,\,V_2)\begin{pmatrix}\Sigma_1^2 & 0\\ 0 & \Sigma_2^2\end{pmatrix}(V_1\,,\,V_2)^T = \tag{15}$$

$$(\overline{V}_1\,,\,\overline{V}_2)\begin{pmatrix}\overline{\Sigma}_1^2 + m\eta^2 I_k & 0\\ 0 & m\eta^2 I_{n-k}\end{pmatrix}(\overline{V}_1\,,\,\overline{V}_2)^T,$$

where I_k and I_{n-k} are identity matrices. From the SVD of H we can then estimate k as the numerical rank of H with respect to the threshold $m^{1/2}\eta$. Furthermore we can use the subspace $\mathcal{R}(V_1)$ as an estimate of $\overline{\mathcal{S}}$ (see, e.g., [18] for results about the quality of this estimate under perturbations).

We now describe several empirical algorithms for computing the estimate $\widehat{H}$; in these algorithms k is always the numerical rank of H. The simplest approach is to compute $\widehat{H}_{\rm ls}$ as a rank-k *least squares estimate* of H, i.e., $\widehat{H}_{\rm ls}$ is the closest rank-k matrix to H in the 2-norm (and the Frobenius norm),

$$\begin{aligned} &\widehat{H}_{\rm ls} = \mathrm{argmin}_{\widehat{H}} \|H - \widehat{H}\|_2 \\ &\quad \text{s.t.} \quad \mathrm{rank}(\widehat{H}) = k. \end{aligned} \tag{16}$$

The Eckart-Young-Mirsky theorem (see [2, Thm. 1.2.3] or [16, Thm. 2.5.3]) expresses this solution in terms of the SVD of H:

$$\widehat{H}_{\rm ls} = U_1\,\Sigma_1\,V_1^T. \tag{17}$$

If desired, it is easy to incorporate the negative "shift" mentioned above. It follows immediately from (15) that

$$\overline{\Sigma}_1^2 = \Sigma_1^2 - m\eta^2 I_k = (I_k - m\eta^2\Sigma_1^{-2})\,\Sigma_1^2,$$

Table 1
Overview of some important gain matrices Φ in the SVD-based methods for the white-noise case.

Estimate	Gain matrix Φ
LS	I_k
MLS	$\left(I_k - m\eta^2\,\Sigma_1^{-2}\right)^{1/2}$
MV	$I_k - m\eta^2\,\Sigma_1^{-2}$
TDC	$\left(I_k - m\eta^2\,\Sigma_1^{-2}\right)\cdot \left(I_k - m\eta^2(1-\lambda)\Sigma_1^{-2}\right)^{-1}$

which lead Van Huffel [52] to defined a *modified least squares estimate*:

$$\begin{aligned} \widehat{H}_{\rm mls} &= U_1\,\Phi_{\rm mls}\,\Sigma_1 V_1^T \quad \text{with} \\ \Phi_{\rm mls} &= (I_k - m\eta^2\Sigma_1^{-2})^{1/2}. \end{aligned} \tag{18}$$

The estimate $\hat{s}$ from this approach is an empirical least-squares estimate of $\bar{s}$.

A number of alternative estimates have been proposed. For example, De Moor [7] introduced the *minimum variance estimate* $\widehat{H}_{\rm mv}$ which satisfies the criterion

$$\begin{aligned} &\widehat{H}_{\rm mv} = \text{argmin}_{\widehat{H}} \|\overline{H} - \widehat{H}\|_{\rm F} \\ &\quad \text{s.t.} \quad \widehat{H} = HW_{\rm mv} \end{aligned} \tag{19}$$

for some matrix $W_{\rm mv}$, and he showed (see our Appendix) that this estimate is given by

$$\begin{aligned} \widehat{H}_{\rm mv} &= U_1\,\Phi_{\rm mv}\,\Sigma_1\,V_1^T \quad \text{with} \\ \Phi_{\rm mv} &= I_k - m\eta^2\Sigma_1^{-2}. \end{aligned} \tag{20}$$

Ephraim and Van Trees [12] defined a *time-domain constraint estimate* which, in our notation, takes the form $\widehat{H}_{\rm tdc} = HW_{\rm tdc}$, where $W_{\rm tdc}$ satisfies the criterion

$$\begin{aligned} &W_{\rm tdc} = \text{argmin}_{\widehat{W}} \|\overline{H} - \overline{H}W\|_{\rm F} \\ &\quad \text{s.t.} \quad \|W\|_{\rm F} \le \alpha\sqrt{m}, \end{aligned} \tag{21}$$

in which α is a user-specified positive parameter. If the constraint is active, then the matrix $W_{\rm tdc}$ is given by the Wiener solution[4]

$$W_{\rm tdc} = \overline{V}_1\,\overline{\Sigma}_1^2(\overline{\Sigma}_1^2 + \lambda\,m\eta^2 I_k)^{-1}\overline{V}_1^T$$

where λ is the Lagrange parameter for the inequality constraint in (21). If we use (15) then we can write the TDC estimate in terms of the SVD of H as

$$\begin{aligned} \widehat{H}_{\rm tdc} &= U_1\,\Phi_{\rm tdc}\,\Sigma_1\,V_1^T \quad \text{with} \\ \Phi_{\rm tdc} &= \left(I_k - m\eta^2\Sigma_1^{-2}\right)\cdot \\ &\qquad \left(I_k - m\eta^2(1-\lambda)\Sigma_1^{-2}\right)^{-1}. \end{aligned} \tag{22}$$

[4]In the regularization literature, $W_{\rm tdc}$ is known as a Tikhonov solution [19].

This relation is derived in our Appendix. If the constraint is inactive then $\lambda = 0$ and we obtain the LS solution. Note that we obtain the MV solution for $\lambda = 1$.

All these algorithms can be written in a unified formulation as

$$\widehat{H}_{\rm svd} = U_1\,\Phi\,\Sigma_1\,V_1^T \tag{23}$$

where Φ is a diagonal matrix, called the *gain matrix*, determined by the optimality criterion; see Table 1. Other choices of Φ are discussed in [48]. The corresponding Matlab code for the MV estimate is

```
[U,S,V] = svd(H,0);
k = length(diag(S) > sqrt(m)*eta);
Phi = eye(k) - m*eta^2*inv(S(1:k,1:k)^2);
Hhat = U(:,1:k)*Phi*S(1:k,1:k)*V(:,1:k)';
```

with the codes for the other estimates being almost similar (only the expression for `Phi` changes).

A few practical remarks are in order here. The MLS, MV and TDC methods require knowledge about the noise's variance η^2; good estimates of this quantity can be obtained from samples of the noise e in the speech pauses. The thresholds used in all our Matlab templates (here, $\tau = \sqrt{m}\,\eta$) are the ones determined by the theory. In practise, we advice the inclusion of a "safety factor," say, $\sqrt{2}$ or 2, in order to ensure that k is an underestimate (because overestimates included noisy components). However, since this factor is somewhat problem dependent, it is not included in our templates.

We note that equation (23) can also be written as

$$\widehat{H}_{\rm svd} = H\,W_\Phi, \quad W_\Phi = V\begin{pmatrix}\Phi & 0\\ 0 & 0\end{pmatrix}V^T \tag{24}$$

where W_Φ is a symmetric matrix which takes care of both the truncation at k, and the modification of the singular values (W_Φ is a projection matrix in the LS case only). Using this formulation, we immediately see that the estimate $\hat{s}$ (8) takes the simple form

$$\hat{s} = W_\Phi\,H(\ell,:)^T = W_\Phi\,s, \tag{25}$$

where s is a arbitrary length-n signal vector. This approach is useful when the signal is quasi-stationary for longer periods, and the same filter, determined by W_Φ, can be used over these periods (or in an exponential window approach).

4. RANK-REVEALING TRIANGULAR DECOMPOSITIONS

In real-time signal processing applications the computational work in the SVD-based algorithms, both in computing and updating the decompositions, may be too large. Rank-revealing triangular decompositions are computationally attractive alternatives which are faster to compute than the SVD, because they involve an initial factorization that can take advantage of the Hankel structure, and they are also much faster to update than the SVD.

Below we present these decompositions and their use. Our Matlab examples required the UTV TOOLS package [13] and, for the VSV decomposition, also the UTV EXPANSION PACK [14]. These packages include software for efficient computation of all the decompositions, as well as software for up- and downdating. The software is designed such that one can either estimate the numerical rank or use a fixed, predetermined value for k.

4.1. UTV Decompositions

Rank-revealing UTV decompositions were introduced in the early 90es by Stewart [44], [45] as alternatives to the SVD, and take the form (referred to as URV and ULV, respectively):

$$H = U_R \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix} V_R^T, \tag{26}$$

$$H = U_L \begin{pmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{pmatrix} V_L^T, \tag{27}$$

where $R_{11}, L_{11} \in \mathbb{R}^{k\times k}$. The four "outer" matrices $U_L, U_R \in \mathbb{R}^{m\times n}$, and $V_L, V_R \in \mathbb{R}^{n\times n}$ have n orthonormal columns, and the numerical rank[5] of H is revealed in the middle $n \times n$ triangular matrices:

$$\sigma_i(R_{11}) \approx \sigma_i(L_{11}) \approx \sigma_i(H), \quad i = 1, \ldots, k$$

$$\left\| \begin{pmatrix} R_{12} \\ R_{22} \end{pmatrix} \right\|_{\mathrm{F}} \approx \|(\, L_{21} \, , \, R_{22} \,)\|_{\mathrm{F}} \approx \sigma_{k+1}(H).$$

In our applications we assume that there is a well-defined gap between σ_k and σ_{k+1}. The more work one is willing to spend in the UTV algorithms, the smaller the norm of the off-diagonal blocks R_{12} and L_{21}.

In addition to information about numerical rank, the UTV decompositions also provide approximations to the SVD subspaces, cf. §3.3 in [19]. For example, if $V_{R1} = V_R(:, 1\colon k)$ then the subspace angle $\angle(V_1, V_{R1})$ between the ranges of V_1 (in the SVD) and V_{R1} (in the URV decomposition) satisfies

$$\sin\angle(V_1, V_{R1}) \le \frac{\sigma_k(R_{11}) \, \|R_{12}\|_2}{\sigma_k(R_{11})^2 - \|R_{22}\|_2^2}.$$

The similar result for $V_{L1} = V_L(:, 1\colon k)$ in the ULV decomposition takes the form

$$\sin\angle(V_1, V_{L1}) \le \frac{\|L_{21}\|_2 \, \|L_{22}\|_2}{\sigma_k(L_{11})^2 - \|L_{22}\|_2^2}.$$

[5] The case where H is exactly rank deficient, for which the submatrices R_{12}, R_{22}, L_{21} and L_{22} are zero, was treated much earlier by Golub [15] in 1965.

Table 2
Symmetric gain matrices Ψ for UTV and VSV (for the white noise case), using the notation T_{11} for either R_{11}, L_{11} or S_{11}.

Estimate	Gain matrix Ψ
LS	I_k
MV	$I_k - m\eta^2 T_{11}^{-1} T_{11}^{-T}$
TDC	$\left(I_k - m\eta^2 T_{11}^{-1} T_{11}^{-T}\right) \cdot \left(I_k - m\eta^2 (1-\lambda) T_{11}^{-1} T_{11}^{-T}\right)^{-1}$

We see that the smaller the norm of R_{12} and L_{21}, the smaller the angle. The ULV decomposition can be expected to give better approximations to the signal subspace $\mathcal{R}(V_1)$ than URV when there is a well-defined gap between σ_k and σ_{k+1}, due to the factors $\sigma_k(R_{11} \approx \sigma_k$ and $\|L_{22}\|_2 \approx \sigma_{k+1}$ in these bounds.

For special cases where the off-diagonal blocks R_{12} and L_{21} are zero, and under the assumption that $\sigma_k(T_{11}) > \|T_{22}\|_2$ – in which case $\mathcal{R}(V_{T1}) = \mathcal{R}(V_1)$ – we can derive explicit formulas for the estimators from Section 3. For example, the least squares estimates are obtained by simply neglecting the bottom block T_{22} – similar to neglecting the block Σ_2 in the SVD approach. The MV and TDC estimates are derived in the Appendix.

In practise, the off-diagonal blocks are not zero but have small norm, and therefore it is reasonable to also neglect these blocks. In general, our UTV-based estimates thus take the form

$$\widehat{H}_{\mathrm{utv}} = U_T \begin{pmatrix} T_{11}\,\Psi & 0 \\ 0 & 0 \end{pmatrix} V_T^T, \tag{28}$$

where $T = R, L$ and where the symmetric gain matrix Ψ is given in Table 2. The MV and TDC formulations, which are derived by replacing the matrix in Σ_1^2 in Table 1 with $T_{11}^T T_{11}$, were originally presented in [23]; there is no estimate that corresponds to MLS. We emphasize again that these estimators only satisfy the underlying criterion when the off-diagonal block is zero.

In analogy with the SVD-based methods, we can use the alternative formulations

$$\widehat{H}_{\mathrm{urv}} = H\, W_{R,\Psi}, \quad \widetilde{H}_{\mathrm{ulv}} = H\, W_{L,\Psi} \tag{29}$$

with the symmetric matrix $W_{T,\Psi}$ given by

$$W_{T,\Psi} = V_T \begin{pmatrix} \Psi & 0 \\ 0 & 0 \end{pmatrix} V_T^T. \tag{30}$$

The two estimates $\widehat{H}_{\mathrm{ulv}}$ and $\widetilde{H}_{\mathrm{ulv}}$ are not identical; they differ by $U_L(:, k{+}1\colon n)\, L_{21}\, V_L(:, 1\colon k)^T$ whose norm $\|L_{21}\|_2$ is small.

The Matlab code for the ULV case with high rank (i.e., $k \approx n$) takes the form

```
[k,L,V] = hulv(H,eta);
Ik = eye(k);
Psi = Ik - m*eta^2*...
      L(1:k,1:k)\Ik/L(1:k,1:k)';
Hhat = H*V(:,1:k)*Psi*V(:,1:k)';
```

An alternative code that requires more storage for `U` has the form

```
[k,L,V,U] = hulv(H,eta);
Psi = Ik - m*eta^2*...
      L(1:k,1:k)\Ik/L(1:k,1:k)';
Hhat = U(:,1:k)*L(1:k,1:k)*Psi*V(:,1:k)';
```

For the ULV case with low rank ($k \ll n$) change `hulv` to `lulv`, and for the URV cases change `_ulv` to `_urv`.

4.2. Symmetric VSV Decompositions

If the signal length N is odd and we use $m = n$ (ignoring the condition $m \geq n+k$), then the square Hankel matrices H and E are symmetric. It is possible to utilize this property in both the SVD and the UTV approach.

In the former case, we can use that a symmetric matrix has the eigenvalue decomposition

$$H = V\,\Lambda\,V^T$$

with real eigenvalues in Λ and orthonormal eigenvectors in V, and thus the SVD of H can be written as

$$H = V\,D\,|\Lambda|\,V^T, \qquad D = \mathrm{diag}(\mathrm{sign}(\lambda_i)).$$

This well-known result essentially halves the work in computing the SVD. The remaining parts of the algorithm are the same, using $|\Lambda|$ for Σ.

In the case of triangular decompositions, a symmetric matrix has a symmetric rank-revealing VSV decomposition of the form

$$H = V_S \begin{pmatrix} S_{11} & S_{12} \\ S_{12}^T & S_{22} \end{pmatrix} V_S^T \tag{31}$$

where $V_S \in \mathbb{R}^{n\times n}$ is orthogonal, and $S_{11} \in \mathbb{R}^{k\times k}$ and S_{22} are symmetric. The decomposition is rank revealing in the sense that the numerical rank is revealed in the "middle" $n \times n$ symmetric matrix:

$$\sigma_i(S_{11}) \approx \sigma_i(H), \qquad i = 1,\ldots,k$$

$$\left\| \begin{pmatrix} S_{12} \\ S_{22} \end{pmatrix} \right\|_{\mathrm{F}} \approx \sigma_{k+1}(H).$$

The symmetric rank-revealing VSV decomposition was originally proposed by Luk and Qiao [35], and it was further developed in [24].

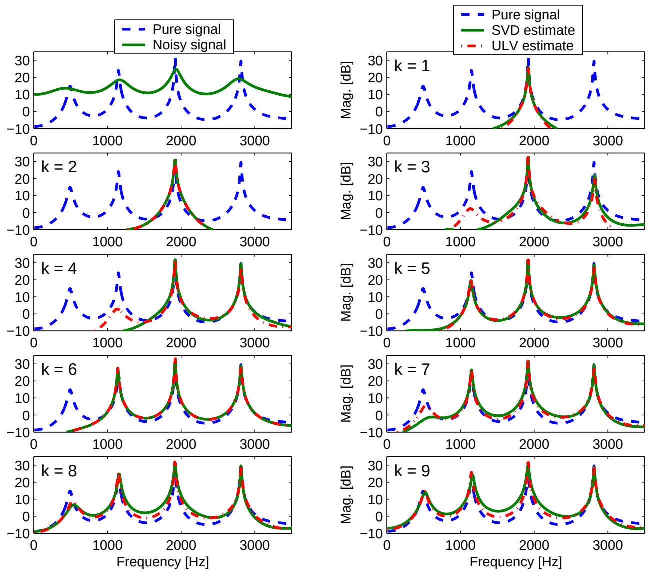

Figure 2. Example with a sum-of-sines clean signal for which $\overline{H}$ has rank 8, and additive white noise with SNR 0 dB. Top left: LPC spectra for the clean and noisy signals. Other plots: LPC spectral for the SVD and ULV LS-estimates with truncation parameter $k = 1,\ldots,9$.

The VSV-based matrix estimate is then given by

$$\widehat{H}_{\mathrm{vsv}} = V_S \begin{pmatrix} S_{11}\Psi & 0 \\ 0 & 0 \end{pmatrix} V_S^T, \tag{32}$$

in which the gain matrix Ψ is computed from Table 2 with T_{11} replaced by the symmetric matrix S_{11}. Again, these expressions are derived under the assumption that $S_{12} = 0$; in practice the norm of this block is small.

The algorithms in [24] for computing VSV decompositions return a factorization of S which, in the indefinite case, takes the form

$$S = T^T\,\Omega\,T,$$

where T is upper or lower triangular, and $\Omega = \mathrm{diag}(\pm 1)$. Below is Matlab code for the high-rank case ($k \approx n$):

```
[k,R,Omega,V] = hvsvid_R(A,eta);
Ik = eye(k);
M = R(1:k,1:k)'\Ik/R(1:k,1:k);
M = Omega(1:k,1:k)*M*Omega(1:k,1:k);
Psi = Ik - R(1:k,1:k)\M/R(1:k,1:k)';
Hhat = V(:,1:k)*S(1:k,1:k)*Psi*V(:,1:k)';
```

5. WHITE NOISE EXAMPLE

We start with an illustration of the noise reduction for the white-noise case by means of SVD and ULV, using an artificially generated clean signal:

$$\bar{s}_i = \sin(0.4i) + 2\sin(0.9i) + 4\sin(1.7i) + 3\sin(2.6i)$$

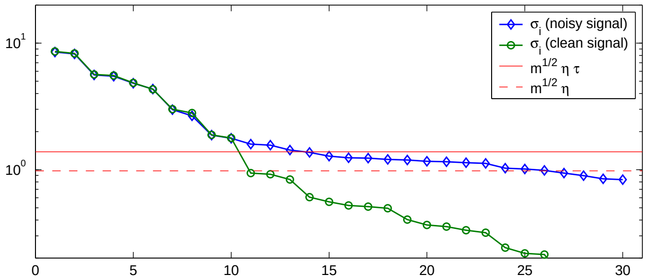

Figure 3. The singular values of the Hankel matrices $\overline{H}$ (clean signal) and H (noisy signal). The solid horizontal line is the "safeguarded" threshold $\sqrt{2}\,m^{1/2}\eta$; the numerical rank with respect to this threshold is $k = 13$.

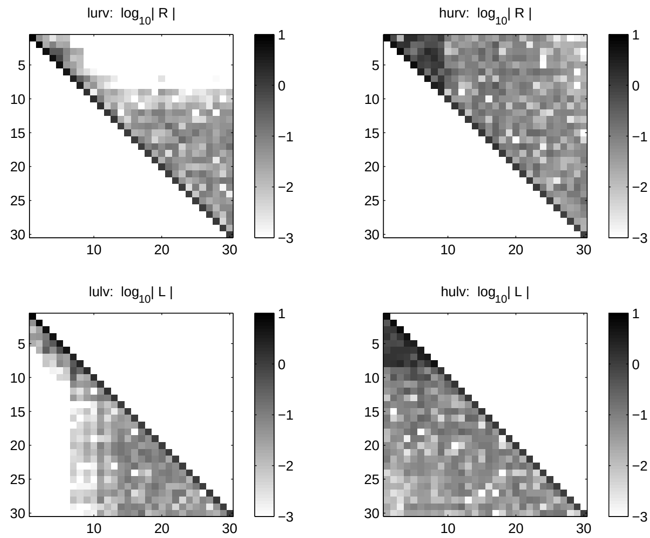

Figure 4. The large and small singular values are reflected in the size of the elements in the matrices R and L from the URV and ULV decompositions. The triangular matrices from the `lurv` and `lulv` algorithms (left plots) are closer to block diagonal form than those from the `hurv` and `hulv` algorithms (right plots).

for $i = 1 \ldots, N$. This signal satisfies the subspace assumption, and the corresponding clean data matrix $\overline{H}$ has rank 8.

We add white noise with SNR $= 0$ dB, and compute SVD and ULV LS-estimates for $k = 1, \ldots, 9$. Figure 2 shows for LPC spectra for each signal, and we see that the two algorithms produce very similar results.

This example illustrates that as k increases we include an increasing number of spectral components, and this occurs in the order of decreasing energy of these components. It is precisely this behavior of the subspace algorithms that make them so powerful for signals that (approximately) admit the subspace model.

We now turn to the speech signal from Fig.1, recalling that this signal does not satisfy the subspace assumption exactly. Figure 3 shows the singular values of the two Hankel matrices $\overline{H}$ and H associated with the clean and noisy signals. We see that the larger singular values of H are quite similar to those of $\overline{H}$, i.e., they are not affected very much by the noise – while the smaller singular values of H tend to level off around $\sqrt{m}\eta$, which is the variance of the noise. Figure 3 also shows our "safeguarded" threshold $\sqrt{2}\sqrt{m}\eta$ for the truncation parameter, leading to the choice $k = 13$ for this particular realization of the noise.

The rank-revealing UTV algorithms are designed such that they reveal the large and small singular values of H in the triangular matrices R and L, and Fig. 4 shows a clear grading of the size of the nonzero elements in these matrices. The particular structure of the nonzero elements in R and L depends on the algorithm used to compute the decomposition. We see that the "low-rank versions" `lurv` and `lulv` tend to produce triangular matrices whose off-diagonal blocks R_{12} and L_{21} have smaller elements than those from the "high-rank versions" `hurv` and `hulv` (see [13] for more details about these algorithms).

Next we illustrate the performance of the SVD- and ULV-based algorithms using the minimum-variance (MV) estimates. The top plot in Fig. 5 shows the LPC spectra for the clean and noisy signals – in the clean signal we see four distinct formants, while only two formants are above the noise level in the noisy signal.

The middle and bottom plots in Fig. 5 show the spectra for the MV estimates using the SVD and ULV algorithms with truncation parameters $k = 8$ and $k = 16$, respectively. Note that the SVD- and ULV-estimates have almost identical spectra for a fixed k, illustrating the usefulness of the more efficient ULV algorithm. For $k = 8$ the two largest formants are well reconstructed; but k is too low to allow us to capture all four formants. For $k = 16$ all four formants are reconstructed satisfactorily, while a larger value of k leads to the inclusion of too much noise. This illustrates the importance of choosing the correct truncation parameter. The clean and estimated signals are compared in Fig. 6.

6. GENERAL NOISE

We now turn to the case of more general noise whose covariance matrix C_e is no longer a scaled identity matrix. We still assume that the noise and the pure signal are uncorrelated and that C_e has full rank. Let C_e have the Cholesky factorization

$$C_e = R_e^T R_e, \tag{33}$$

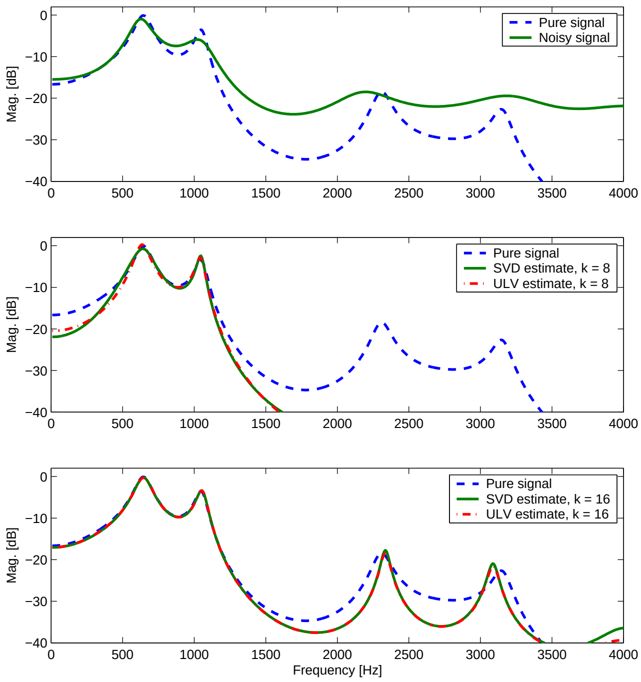

Figure 5. LPC spectra of the signals in the white-noise example, using SVD- and ULV-based MV estimates. Top: clean and noisy signals. Middle and bottom: estimates; both SNRs are 12.5 dB for $k = 8$ and 13.8 dB for $k = 16$.

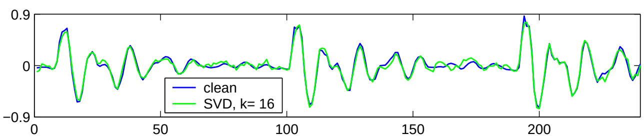

Figure 6. Comparison of the clean signal and the SVD-based MV estimate for $i = 16$.

where R_e is an upper triangular matrix of full rank. Then the standard approach is to consider the transformed signal $R_e^{-T} s$ whose covariance matrix is given by

$$\begin{aligned} &\mathcal{E}(R_e^{-T} s\, s^T R_e^{-1}) \\ &\quad = \quad R_e^{-T} C_s R_e^{-1} = R_e^{-T} C_{\bar{s}} R_e^{-1} + I_n, \end{aligned} \tag{34}$$

showing that the transformed signal consists of a transformed pure signal plus additive white noise with unit variance. Hence the name *prewhitening* for this process. Clearly, we can apply all the methods from the previous section to this transformed signal, followed by a back-transformation involving multiplication with R_e^T.

Turning to practical algorithms based on the cross-product matrix estimates for the covariance matrices, our assumptions are now

$$\operatorname{rank}(E) = n \qquad \text{and} \qquad \overline{H}^T E = 0.$$

Since E has full rank, we can compute an orthogonal factorization $E = Q\,R$ in which Q has orthonormal columns and R is nonsingular. For example, if we use a QR factorization then R is a Cholesky factor of $E^T E$, and $m^{-1/2} R$ estimates R_e above. We introduce the transformed signal $z_{\mathrm{qr}} = R^{-T} s$ whose covariance matrix is estimated by

$$\tfrac{1}{m} R^{-T} H^T H\, R^{-1} = \tfrac{1}{m} R^{-T} \overline{H}^T \overline{H}\, R^{-1} + \tfrac{1}{m} I_n,$$

showing that the prewhitened signal z_{qr} – similar to above – consists of a transformed pure signal plus additive white noise with variance m^{-1}. Again we can apply any of the methods from the previous section to the transformed signal z_{qr}, represented by the matrix $Z_{\mathrm{qr}} = H\,R^{-1}$, followed by a back-transformation with R^T.

The complete model algorithm for treating full-rank non-white noise thus consists of the following steps. First compute the QR factorization $E = Q\,R$, then form the prewhitened matrix $Z_{\mathrm{qr}} = H\,R^{-1}$ and compute its SVD $Z_{\mathrm{qr}} = U\,\Sigma\,V^T$. Then compute the “filtered” matrix $\widehat{Z}_{\mathrm{qr}} = Z_{\mathrm{qr}}\,W_\Phi$ with the gain matrix Φ from Table 1 using $m\eta^2 = 1$. Finally compute the dewhitened matrix $\widehat{H}_{\mathrm{qr}} = \widehat{Z}_{\mathrm{qr}}\,R$ and extract the filtered signal. For example, for the MV estimate this is done by the following Matlab code

```
[Q,R] = qr(E,0);
[U,S,V] = svd(H/R,0);
k = length(diag(S) > 1/sqrt(m));
Phi = eye(k) - inv(S(1:k,1:k))^2;
Hhat = U(:,1:k)*Phi*S(1:k,1:k)...
       *V(:,1:k)'*R;
```

6.1. GSVD Methods

There is a more elegant version of the above algorithm which avoids the explicit pre- and dewhitening steps, and which can be extended to a rank deficient E, cf. §7. It can be formulated both in terms of the covariance matrices and their cross-product estimates.

Consider first the covariance matrix approach [25], [26], which is based on the generalized eigenvalue decomposition of $C_{\bar{s}}$ and C_e:

$$C_{\bar{s}} = \overline{X}\,\bar{\Lambda}\,\overline{X}^T, \quad C_e = \overline{X}\,\overline{X}^T,$$

where $\bar{\Lambda} = \mathrm{diag}(\bar{\lambda}_1, \ldots, \bar{\lambda}_n)$ and $\overline{X}$ is a nonsingular matrix[6] (see, e.g., §8.7 in [16]). If we partition $\overline{X} = (\,\overline{X}_1\,,\,\overline{X}_2\,)$ with $\overline{X}_1 \in \mathbb{R}^{n\times k}$ then the pure signal subspace satisfies $\overline{S} = \mathcal{R}(\overline{X}_1)$. Moreover

$$C_s = C_{\bar{s}} + C_e = \overline{X}\,(\bar{\Lambda} + I_n)\,\overline{X}^T,$$

showing that we can perfectly reconstruct $C_{\bar{s}}$ (similar to the white noise case) by subtracting 1 from the k largest generalized eigenvalues of C_s.

As demonstrated in [30], we can turn the above into a working algorithm by means of the generalized SVD (GSVD) of H and E, given by

$$H = U_H\,\Gamma\,X^T, \qquad E = U_E\,\Delta\,X^T. \tag{35}$$

If E has full rank then $X \in \mathbb{R}^{n\times n}$ is nonsingular. Moreover, $U_H, U_E \in \mathbb{R}^{m\times n}$ have orthonormal columns, and $\Gamma, \Delta \in \mathbb{R}^{n\times n}$ are diagonal matrices

$$\Gamma = \mathrm{diag}(\gamma_1, \ldots, \gamma_n), \quad \Delta = \mathrm{diag}(\delta_1, \ldots, \delta_n)$$

satisfying $\Gamma^2 + \Delta^2 = I$ (see, e.g., [2, §4.2]). In the QR-based algorithm described above, we now replace the QR factorization of E with the factorization $E = U_E\,(\Delta\,X^T)$, leading to a matrix Z_{gsvd} given by

$$Z_{\mathrm{gsvd}} = H\,(\Delta\,X^T)^{-1} = U_H\,(\Gamma\,\Delta^{-1}), \tag{36}$$

which is the SVD of Z_{gsvd} expressed in terms of GSVD factors. The corresponding signal $z_{\mathrm{gsvd}} = (\Delta\,X^T)^{-T}s = (X\,\Delta)^{-1}s$ consists of the transformed pure signal $(X\,\Delta)^{-1}\bar{s}$ plus additive white noise with variance m^{-1}. Also, the pure signal subspace is spanned by the first k columns of X, i.e., $\overline{S} = \mathcal{R}(X(:,1{:}\,k))$.

Let Γ_1 and Δ_1 denote the leading $k\times k$ submatrices of Γ and Δ. Then the filtered and dewhitened matrix $\widehat{H}_{\mathrm{gsvd}}$ takes the form

$$\widehat{H}_{\mathrm{gsvd}} = U_H\,\Gamma \begin{pmatrix} \Phi & 0 \\ 0 & 0 \end{pmatrix} X^T = H\,Y_\Phi \tag{37}$$

with

$$Y_\Phi = X^{-T}\begin{pmatrix} \Phi & 0 \\ 0 & 0 \end{pmatrix} X^T, \tag{38}$$

where again Φ is from Table 1 with $\Sigma_1 = \Gamma_1\,\Delta_1^{-1} = \Gamma_1\,(I - \Gamma_1^2)^{-1/2}$ and $m\eta^2 = 1$. Thus we can compute the filtered signal either by averaging along the antidiagonals of $\widehat{H}_{\mathrm{gsvd}}$ or as

$$\hat{s}_{\mathrm{gsvd}} = Y_\Phi^T s = X(:,1{:}\,k)\,(\,\Phi\,,\,0\,)\,X^{-1}s.$$

The Matlab code for MV case takes the form

```
[U,V,X,Gamma,Delta] = gsvd(H,E,0);
S = Gamma/Delta;
k = length(diag(S) > 1);
Phi = eye(k) - inv(S(1:k,1:k))^2;
Hhat = U(:,1:k)*Gamma(1:k,1:k)...
       *Phi*X(:,1:k)';
```

We note that if we are given (an estimate of) the noise covariance matrix C_e instead of the noise matrix E, then in the GSVD-based algorithm we can replace the matrix E with the Cholesky factor R_e in (33).

6.2. Triangular Decompositions

Just as the URV and ULV decompositions are alternatives to the SVD – with a middle triangular matrix instead of middle diagonal matrix – there are alternatives to the GSVD with middle triangular matrices. They also come in two versions with upper and lower triangular matrices but, as shown in [20], only the version using lower triangular matrices is useful in our applications.

This version is known as the ULLV decomposition of H and E; it was introduced by Luk and Qian [34] and it takes the form

$$H = U_H\,L_H\,L\,V^T, \qquad E = U_E\,L\,V^T, \tag{39}$$

where $L_H, L \in \mathbb{R}^{n\times n}$ are lower triangular, and the three matrices $U_H, U_E \in \mathbb{R}^{m\times n}$ and $V \in \mathbb{R}^{n\times n}$ have orthonormal columns. See [23] for an application of the ULLV decomposition in speech processing.

The prewhitening technique from §6 carries over to the ULLV decomposition. Using the orthogonal decomposition of E in (39) we define the transformed (prewhitened) signal $z_{\mathrm{ullv}} = (L\,V^T)^{-T}s = L^{-T}V^Ts$ whose scaled covariance matrix is estimated by $\frac{1}{m}Z_{\mathrm{ullv}}^T Z_{\mathrm{ullv}}$, in which

$$Z_{\mathrm{ullv}} = H\,(L\,V^T)^{-1} = U_H\,L_H,$$

and we see that the ULLV decomposition automatically provides a ULV decomposition of this matrix. Hence we can use the techniques from §4.1 to obtain the estimate

$$\widehat{Z}_{\mathrm{ullv}} = U_H \begin{pmatrix} L_{H,11}\,\Psi & 0 \\ 0 & 0 \end{pmatrix},$$

[6] The matrix $\overline{X}$ is not orthogonal; it is chosen such that the columns $\bar{\xi}_i$ of $\overline{X}^{-T}$ satisfy $C_{\bar{s}}\,\bar{\xi}_i = \lambda_i\,C_e\,\bar{\xi}_i$ for $i = 1, \ldots, n$, i.e., $(\bar{\lambda}_i, \bar{\xi}_i)$ are the generalized eigenpairs of $(C_{\bar{s}}, C_e)$.

where $L_{H,11}$ denotes the leading $k \times k$ submatrix of L_H. This leads to the ULLV-based estimate

$$\begin{aligned} \widehat{H}_{\text{ullv}} &= \widehat{Z}_{\text{ullv}} L\, V^T \\ &= U_H \begin{pmatrix} L_{H,11}\, \Psi & 0 \\ 0 & 0 \end{pmatrix} L\, V^T. \end{aligned} \tag{40}$$

The alternative version takes the form

$$\begin{aligned} \widetilde{H}_{\text{ullv}} &= H\, Y_\Psi \quad \text{with} \\ Y_\Psi &= V\, L^{-1} \begin{pmatrix} \Psi & 0 \\ 0 & 0 \end{pmatrix} L\, V^T, \end{aligned} \tag{41}$$

and the gain matrix Ψ is given by the expressions in Table 2 with T_{11} replaced by $L_{H,11}$ and $m\eta^2 = 1$. The Matlab code for the MV estimate is:

```
[k,LH,L,V,UH] = ullv(H,E,1);
Ik = eye(k);
Psi = Ik - LH(1:k,1:k)\Ik/LH(1:k,1:k)';
Hhat = UH(:,1:k)*LH(1:k,1:k)...
       *Psi*L(1:k,1:k)*V(:,1:k)';
```

Similar to the GSVD algorithm, we can replace E by the Cholesky factor R_e of the noise covariance matrix in (33), if it is available.

6.3. Colored Noise Example

We now switch to the colored noise (the wind signal), and the top plot in Fig. 7 shows the power spectra for the pure and noisy signals, together with the power spectrum for the noise signal which is clearly nonwhite. The middle plot shows the power spectra for the MV-estimates using the GSVD and ULLV algorithms with $k = 15$; the corresponding SNRs are 12.1 dB and 11.4 dB. The GSVD estimate is superior to the ULLV estimate, but both give a satisfactory reduction of the noise in the frequency ranges between and outside the formants.

The bottom plot in Fig. 7 illustrates the performance of the SVD and ULV algorithms applied to this signal (i.e., there is no preconditioning). Clearly, the implicit white-noise assumption is not correct and the estimates are inferior to those using the GSVD and ULLV algorithms because the SVD and ULV algorithms mistake some components of the colored noise for signal.

7. RANK DEFICIENT NOISE

Not all noise signals lead to a full-rank noise matrix E; for example, narrow-band signals often lead to an E that is (numerically) rank deficient. In this case, we may think of the noise as an interfering signal that we need to suppress.

When E is rank deficient, the above GSVD- and ULLV-based methods do not apply because Δ and L become rank deficient. In [22] we extended these algorithms to the rank-deficient case; we summarize the

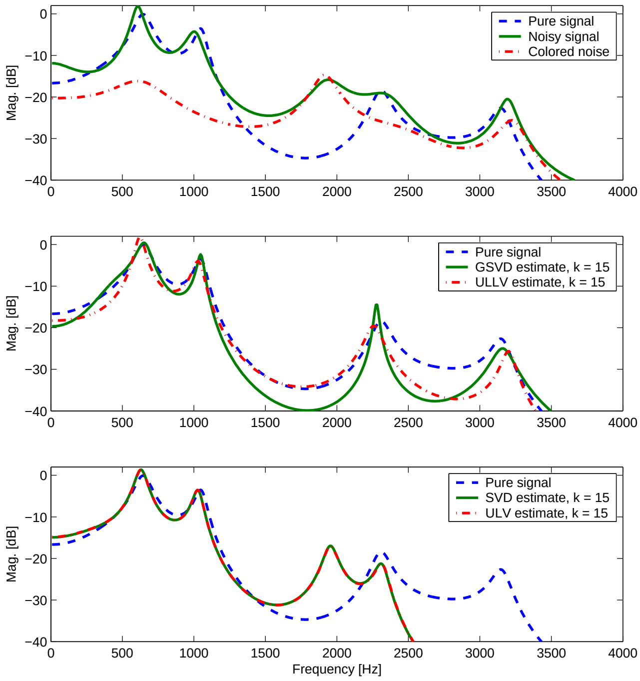

Figure 7. LPC spectra of the signals in the colored-noise example, using the MV estimates. Top: clean and noisy signals together with the noise signal. Middle: GSVD and ULLV estimates; the SNRs are 12.1 dB and 11.4 dB. Bottom: SVD and ULV estimates (both SNRs are 11.4 dB). Without knowledge about the noise, the SVD and ULV methods mistake some components of the colored noise for a signal.

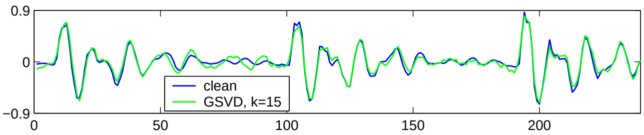

Figure 8. Comparison of the clean signal and the GSVD-based MV estimate for $i = 15$.

algorithms here, and refer to the paper for the – quite technical – details.

The GSVD is not unique in the rank-deficient case, and several formulations appear in the literature. We use the formulation in Matlab, and our algorithms require an initial rank-revealing QR factorization of E of the form

$$E = Q\,R, \quad R \in \mathbb{R}^{p\times n},$$

where R is upper trapezoidal and $p = \mathrm{rank}(E)$. Then we use a GSVD of (H, R) of the form

$$H = U_H \begin{pmatrix} \Gamma & 0 \\ 0 & I_o \end{pmatrix} X^T \tag{42}$$

$$R = U_R\,(\,\Delta\,,\,0\,)\,X^T, \tag{43}$$

where Γ and Δ are $p\times p$ and diagonal, and I_o is the identity matrix of order $n-p$. Moreover, $U_H \in \mathbb{R}^{m\times n}$ and $U_R \in \mathbb{R}^{p\times p}$ have orthonormal columns, and $X \in \mathbb{R}^{n\times n}$ is nonsingular.

The basic idea in our algorithm is to realize that there is no noise in the subspace $\mathcal{R}(X(:, p{+}1\colon n))$ spanned by the last $n-k$ columns of X, and therefore any component of the noisy signal s in this subspace should not be filtered. The filtering should only take place in the subspace $\mathcal{R}(X(:, 1\colon p)$. Note that the vectors in these two subspaces are not orthogonal; as shown in [20], orthogonal subspaces are inferior to the bases $X(:, 1\colon p$ and $X(:, p{+}1\colon n)$.

Again let Γ_1 and Δ_1 denote the leading $k\times k$ submatrices of Γ and Δ. Then the GSVD-based estimate takes the form

$$\widehat{H}_{\mathrm{gsvd}} = U_H \begin{pmatrix} \Gamma & 0 \\ 0 & I_o \end{pmatrix} \left(\begin{array}{cc|c} \Phi & 0 & 0 \\ 0 & 0 & 0 \\ \hline 0 & 0 & I_o \end{array}\right) X^T,$$

where, similar to the full-rank case, the $k\times k$ gain matrix Φ is from Table 1 with $\Sigma_1 = \Gamma_1\,\Delta_1^{-1} = \Gamma_1\,(I-\Gamma_1^2)^{-1/2}$ and $m\eta^2 = 1$.

The corresponding Matlab code for the MV estimate, which requires UTV Tools for the rank-revealing QR factorization `hrrqr`, takes the form (where `thr` is the threshold for the rank decision in E):

```
thr = 1e-12*norm(E,'fro');
[Q,R] = hrrqr(E,thr);
[UH,UR,X,Gamma,Delta] = gsvd(H,R);
S = Gamma/Delta;
k = length(diag(S) > 1);;
i = 1:k; j = p+1:n;
Phi = eye(k) - inv(S(1:k,1:k))^2
Hhat = UH(:,1:k)*Gamma(1:k,1:k)...
       *Phi*X(:,1:k)' + ...
       UH(:,p+1:n)*X(:,p+1:n)';
```

There is also a formulation based on triangular factorizations of H and E. Again assuming that we have first computed the QR factorization of E, this formulation is based on the ULLIV decomposition of (H, R) [20], [22], [36]:

$$H = U_H\,L_H \begin{pmatrix} L & 0 \\ 0 & I_o \end{pmatrix} V^T \tag{44}$$

$$R = U_R\,(\,L\,,\,0\,)\,V^T, \tag{45}$$

in which $U_H \in \mathbb{R}^{m\times n}$, $U_R \in \mathbb{R}^{p\times p}$ and $V \in \mathbb{R}^{n\times n}$ have orthonormal columns, and $L_H \in \mathbb{R}^{n\times n}$ and $L \in \mathbb{R}^{p\times p}$ are lower triangular. The corresponding estimate is given by:

$$\widehat{H}_{\mathrm{ulliv}} = U_H\,L_H \left(\begin{array}{cc|c} \Psi & 0 & 0 \\ 0 & 0 & 0 \\ \hline 0 & 0 & I_o \end{array}\right) \begin{pmatrix} L & 0 \\ 0 & I_o \end{pmatrix} X^T,$$

where Ψ is from Table 2 with T_{11} replaced by $L_{H,11}$, the leading $k\times k$ submatrix of L_H.

The Matlab code requires UTV Tools plus UTV Expansion Pack, and for the MV estimate it takes the form:

```
thr = 1e-12*norm(E,'fro');
[Q,R] = hrrqr(E,thr);
[k,LH,L,V,UH] = ulliv(A,B,1);
Ik = eye(k);
Phi = Ik - LH(1:k,1:k)\Ik/LH(1:k,1:k)';
i = 1:k; j = p+1:n;
Hhat = UH(:,1:k)*LH(1:k,1:k)*Phi*X(:,1:k)' ...
     + UH(:,p+1:n)*LH(p+1:n,p+1:n)*X(:,p+1:n)';
```

8. DYNAMICAL PROCESSING: UP- AND DOWN-DATING

In many applications we are facing a very long signal whose length prevents the "brute-force" use of the above algorithms – for example, the long signal may not be quasi-stationary, and in a real-time application we can only accept a certain small delay caused by the noise reduction algorithm.

A simple approach to obtain real-time processing is to apply the algorithms to short segments whose length is chosen such that the delay is acceptable and such that the signal can be considered quasi-stationary in the duration of the segment. However, this simple block approach can lead to highly undesired modulation effects, due to the fact that the filter changes in each block.

One remedy for this is to impose constraints on how much the filters can change from one block to the next, via imposing a "smoothness constraint" on the basis vectors of the signal subspace from one segment to the next [29]. This approach has proven to reduce the modulation effects considerably, at the expense of a non-negligible increase in computational work.

An alternative approach is to apply the above methods to the signals in a window that either increases in length or has fixed length and "slides" along the given signal. In both cases, we need to recompute the matrix decomposition when the window changes, which leads to the computational problems of up- and downdating.

In the former approach, the task is to compute the factorization of the new, larger Hankel matrix

$$H^{\alpha}_{\rm new} = \begin{pmatrix} \alpha H \\ a^T \end{pmatrix}, \qquad a^T = s(m{+}1\colon N{+}1)$$

where α is a forgetting factor between 0 and 1. The computational problem of efficiently computing the factorization of $H^{\alpha}_{\rm new}$, given the factorization of H, is referred to as *updating*.

In the sliding-window approach, the computational problem becomes that of efficiently computing the factorization of the modified matrix

$$H^{\triangleright}_{\rm new} = \mathcal{H}(\, s(2\colon N{+}1)\,) = \begin{pmatrix} H(2\colon m, :\,) \\ a^T \end{pmatrix}$$

given the factorization of H. We see that this involves a *downdating* step, where the top row is removed from H, followed by an updating step.

Up- and downdating of the SVD is a computationally demanding task which requires of the order n^3 operations when Σ and V are updated, and it involves the solution of nonlinear equations referred to as the secular equations; see [5] and [17] for details. For these reasons, SVD updating is usually considered to be infeasible in real-time applications. This is one of the original motivations for introducing the rank-revealing triangular decompositions, whose up- and downdating requires only of the order n^2 computations.

The details of the up-and downdating algorithms for the UTV, VSV, ULLV and ULLIV decompositions are rather technical; we refer to the packages [13], [14] and the many references in there for details.

9. FIR FILTER INTERPRETATIONS

The behavior and the quality of the rank-revealing matrix factorizations that underly our algorithms is often measured in terms of linear algebra "tools" such as perturbation bound and angles between subspaces. While mathematically well-defined and precise, these "tools" may not give an intuitive interpretation of the performance of the algorithms when applied to digital signals.

The purpose of this section is to demonstrate that we can associate a straight-forward FIR filter interpretation with each algorithm, thus allowing a performance study which is more directly oriented towards the signal processing applications. This section expands the SVD/GSVD-based results from [21] to the methods based on triangular decompositions, and also introduces the new concept of canonical filters.

The FIR filter interpretation is most conveniently explained in connection with the estimate $\hat{s}'$ (7) obtained via averaging along the anti-diagonals of the matrix estimate $\widehat{H}$. This interpretation is based on the fact that multiplication of a vector x by a Hankel matrix $\mathcal{H}(s')$,

$$[\mathcal{H}(s')\, x]_i = \sum_{j=1}^{n} x_j\, s_{i-j-1}, \qquad i = 1, \ldots, m$$

is equivalent to filtering the signal s' with a FIR filter whose coefficients are the elements of x.

9.1. Basic Relations

If $\mathcal{A}(\cdot)$ denotes the averaging operation defined in (8), then for an outer product $vw^T \in \mathbb{R}^{m\times n}$ we have

$$\mathcal{A}(vw^T) = D^{-1}\, \mathcal{H}'(v)\, Jw,$$

where D is the $N \times N$ diagonal matrix

$$D = \mathrm{diag}(1, 2, \ldots, n{-}1, n, \ldots, n, n{-}1, \ldots, 2, 1),$$

$\mathcal{H}'(v)$ is the $N \times n$ Hankel matrix with zero upper and lower "corners"

$$\mathcal{H}'(v) = \begin{pmatrix} 0 & 0 & \cdots & v_1 \\ \vdots & \vdots & & \vdots \\ 0 & v_1 & \cdots & v_{n-1} \\ v_1 & v_2 & \cdots & v_n \\ v_2 & v_3 & \cdots & v_{n+1} \\ \vdots & \vdots & & \vdots \\ v_{m-n+1} & v_{m-n+2} & \cdots & v_m \\ v_{m-n+2} & v_{m-n+3} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ v_m & 0 & \cdots & 0 \end{pmatrix},$$

and J is the $n \times n$ exchange matrix consisting of the columns of the identity matrix in reverse order, cf. [11], [21]. If $V_k = (v_1, \ldots, v_k)$ and $\Phi = \mathrm{diag}(\phi_1, \ldots, \phi_k)$ then it follows from (24) that we can write

$$\widehat{H} = H\, W_\Phi = \sum_{i=1}^{k} (Hv_i)\, \phi_i\, v_i^T,$$

and it follows that

$$\begin{aligned} \hat{s}' &= \mathcal{A}(\widehat{H}) \\ &= \sum_{i=1}^{k} \mathcal{A}\big((Hv_i)\, \phi_i\, v_i^T\big) \\ &= \sum_{i=1}^{k} \phi_i\, D^{-1}\, \mathcal{H}'(Hv_i)\, Jv_i \\ &= D^{-1} \sum_{i=1}^{k} \phi_i\, \mathcal{H}'\big(\mathcal{H}(s')v_i\big)\, (Jv_i). \end{aligned} \qquad (46)$$

The time-varying scaling D takes care of corrections at both ends of the signal.

We conclude that the estimate $\hat{s}'$ essentially consists of a weighted sum of k signals, each one obtained by passing the input signal s' through a *pair of FIR filters* with filter coefficients v_i and Jv_i. Each of these filter pairs corresponds to a single FIR filter of length $2n-1$ whose coefficients are the convolution of v_i and Jv_i; i.e., we can write the filter vector as $c_i = \mathcal{H}'(v_i)\, v_i$. These filters[7] are symmetric and have zero phase.

9.2. SVD/UTV/VSV Filters

We first consider the LS algorithms where the filter matrix is the identity, $\Phi = I_k$ and $\Psi = I_k$, which corresponds to a simple truncation of the SVD, UTV or VSV decomposition. Then $\hat{s}'$ is given by (46) with $\phi = 1$ and with v_i denoting the ith column of any of the matrices V, V_L, V_R or V_S (depending on the decomposition used).

The k individual contributions to $\hat{s}'$ can be judged as follows. If we write $\mathcal{H}'(Hv_i) = \|Hv_i\|_2 \mathcal{H}'(\tilde{v}_i)$ with $\tilde{v}_i = Hv_i \|Hv_i\|_2^{-1}$, then we obtain

$$\|\mathcal{H}'(Hv_i)Jv_i\|_2 \le \|Hv_i\|_2\, \|\mathcal{H}'(\tilde{v}_i)\|_2\, \|Jv_i\|_2,$$

where $\|Jv_i\|_2 = 1$. Moreover,

$$\|\mathcal{H}'(\tilde{v}_i)\|_2 \le \|\mathcal{H}'(\tilde{v}_i)\|_{\mathrm{F}} = n^{1/2}\, \|\tilde{v}_i\|_2 = n^{1/2},$$

and thus, for $i = 1, \ldots, k$,

$$\|\mathcal{H}'(Hv_i)Jv_i\|_2 \le n^{1/2}\, \|Hv_i\|_2. \tag{47}$$

For the SVD algorithm $\|Hv_i\|_2 = \sigma_i$. The UTV and VSV algorithms are designed such that $\|Hv_i\|_2 \approx \sigma_i$. This means that the energy in the output signal of the ith filter branch is bounded by σ_i (or an approximation to σ_i). By truncating the decomposition at k we thus include the k most significant components in the signal, as determined by the filters defined by the vectors v_i.

In the next section, we demonstrate that these filters are typically band-pass filters centered at frequencies for which the signal's power spectrum has large values. Hence, the filters "pick out" the dominating spectral components/bands in the signal; this leads to noise reduction because these components/bands are dominated by the pure signal.

For the other SVD-based algorithms (MLS, MV and TDC), $\Phi \ne I$ is still a diagonal matrix and Eq. (46) still holds. The analysis remains unchanged, except that the ith output signal is multiplied by the weight ϕ_i.

For the other UTV- and VSV-based algorithms (MV and TDC), the filter matrix Ψ is a symmetric $k \times k$ matrix with eigenvalue decomposition

$$\Psi = Y\, M\, Y^T, \tag{48}$$

[7]It is easy to verify that we obtain the same FIR filters if we base our algorithms on Toeplitz matrices.

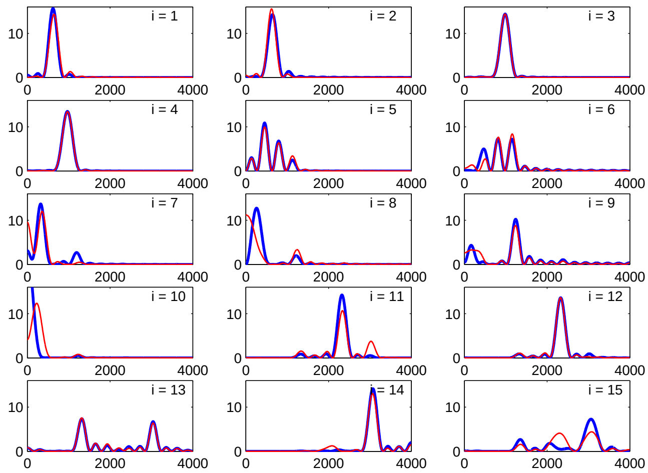

Figure 9. Frequency responses of the combined FIR filters for the SVD algorithm (thick lines) and ULV algorithm (thin lines) applied to the test problem in Fig. 5. Both algorithms compute the MV estimates, and the filters are computed by means of (46) and (49).

in which $M = \mathrm{diag}(\mu_1, \ldots, \mu_k)$ contains the eigenvalues, and the matrix $Y = (y_1, \ldots, y_k)$ contains the orthonormal eigenvectors. Now let $Z = V(:, 1\colon k)\, Y$ denote the $n \times k$ matrix obtained by multiplying the first k columns of V, V_L, V_R or V_S by Y. Then we can write

$$\widehat{H} = H\, Z\, M\, Z^T,$$

which immediately leads to the expression

$$\hat{s}' = D^{-1} \sum_{i=1}^{k} \mu_i\, \mathcal{H}'(Hz_i)Jz_i, \tag{49}$$

where z_i is the ith columns of Z. The FIR filter interpretation described above immediately carries over to this expression: the estimate $\hat{s}'$ is a weighted sum (with weights μ_i) of k signals obtained by passing s' through the filter pairs z_i and Jz_i.

Figure 9 shows the frequency responses for the combined FIR filter pairs associated with the SVD and ULV estimates in Fig. 5. For $i = 1$ through 11, the SVD and ULV filters are very similar in the frequency domain, while some differences show up for $i > 11$. This supports the similarity between the two algorithms that we already noted before.

The first two filters (for $i = 1$ and 2) are bandpass filters that capture the largest formant at 700 Hz, while the next two filters (for $i = 3$ and 4) are bandpass filters that capture the second largest formant at 1.1 kHz. The next six filters (for $i = 5$ through 10) capture more information in the frequency range 0–1500 Hz. The five filters for $i = 11$ through 15 capture the two formants at

2.3 kHz and 3.3 kHz. By adaptively placing bandpass filters at the portions of the signal with high energy, the subspace algorithms are able to extract the most important spectral components of the noisy signal while, at the same time, suppressing the noise in the frequency ranges with less energy.

9.3. GSVD/ULLV Filters

The subspace algorithms for general noise have FIR filter interpretations similar to those for the white-noise algorithms.

To derive the FIR filters for the GSVD algorithms, let $\xi_1, \ldots, \xi_k$ denote the first k columns of the matrix $\Xi = X^{-T}$ in (38), such that

$$\widehat{H}_{\mathrm{gsvd}} = H\, Y_\Phi = H\, \Xi(:\,, 1\colon k)\, \Phi\, X(:\,, 1\colon k)^T.$$

Then it follows from §9.1 that

$$\begin{aligned} \hat{s}' &= \mathcal{A}(H\, Y_\Phi) \\ &= D^{-1} \sum_{i=1}^{k} \phi_i\, \mathcal{H}'\big(\mathcal{H}(s')\xi_i\big)\, Jx_i. \qquad (50) \end{aligned}$$

We see that the coefficients of the ith FIR filter pair ξ_i and Jx_i consist of the elements of the ith columns of $\Xi = X^{-T}$ and X (in reverse order), and the combined filters have coefficients given by the convolution of ξ_i and Jx_i. In contrast to the SVD/UTV/VSV algorithms, these are not zero-phase filters.

In order to obtain bounds similar to (47), we make the reasonable assumption that $\|E\|_2 \le \|H\|_2$. Then it follows from the definition of the GSVD that

$$\|Jx_i\|_2 = \|x_i\|_2 \le \|X\|_2 = \left\| \begin{pmatrix} H \\ E \end{pmatrix} \right\|_2 \le 2\, \|H\|_2.$$

From the definition we also have $\|H\, \xi_i\|_2 = \gamma_i$. Following the same procedure as in the previous section we thus obtain, for $i = 1, \ldots, k$,

$$\|\mathcal{H}'\big(\mathcal{H}(s')\xi_i\big)\, Jx_i\|_2 \le 2n^{1/2}\, \gamma_i\, \|H\|_2. \qquad (51)$$

Similar to before, we thus include the k most significant components in the signal, as determined by the filters defined by the vectors ξ_i and x_i.

For the ULLV algorithm we insert the eigenvalue decomposition of Ψ (48) into (41) to obtain

$$\begin{aligned} \widetilde{H}_{\mathrm{ullv}} &= H\, V L^{-1} \begin{pmatrix} Y M\, Y^T & 0 \\ 0 & 0 \end{pmatrix} L\, V^T \\ &= \sum_{i=1}^{k} (H\, V L^{-1} y_i)\, \mu_i\, (V L^T y_i)^T. \end{aligned}$$

When we insert this result into the expression for $\mathcal{A}(\cdot)$ we immediately obtain

$$\hat{s}' = D^{-1} \sum_{i=1}^{k} \phi_i\, \mathcal{H}'\big(\mathcal{H}(s') V L^{-1} y_i\big)\, J\, V L^T y_i. \qquad (52)$$

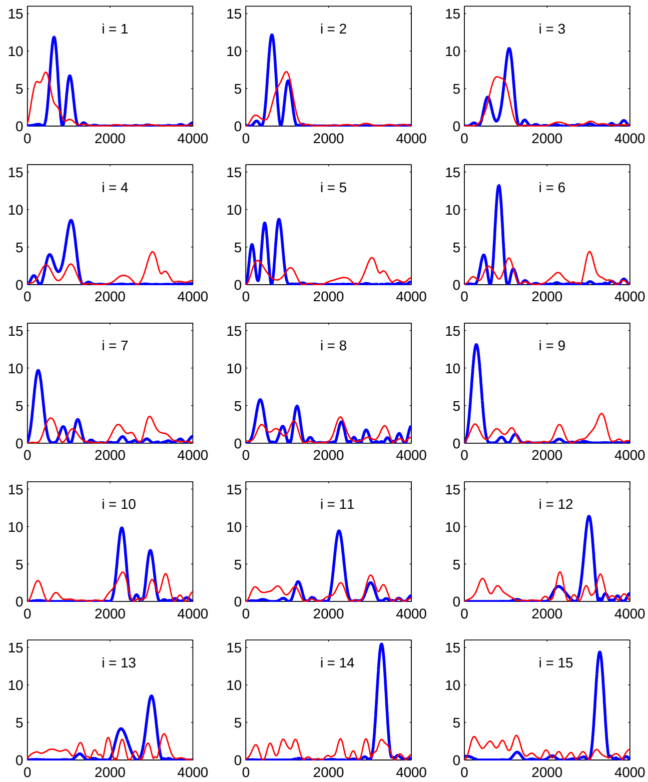

Figure 10. Frequency responses of the FIR filters for the GSVD algorithm (thick lines) and ULLV algorithm (thin lines) applied to the test problem in Fig. 7. Both algorithms compute the MV estimates, and the filters are computed by means of (50) and (52).

and we see that the coefficients of the ith FIR filter pair are the elements of the two vectors $V L^{-1} y_i$ and $J\, V L^T y_i$.

While difficult to immediately interpret, the relations derived in this section allow us to compute the FIR filters and in this way study the performance of the algorithms considered.

10. CANONICAL FILTERS

We shall now present a novel technique, based on the FIR filter interpretation, for comparing the performance of different subspace algorithms. To simplify the presentation, we restrict ourselves to the LS estimation algorithms where $\Phi = \Psi = I_k$. The rank-k matrix estimates take the form

$$\widehat{H} = H\, V_k\, V_k^T,$$

in which V_k denotes the submatrix consisting of the first k columns of V (in the SVD algorithm), U_R or U_L (in the UTV algorithms) or V_S (in the VSV algorithm).

10.1. Theory

The important observation here is that the matrix $\widehat{H}$ is *independent* of the choice of the columns $v_1, \ldots, v_n$ of the matrix V_k, as long as they are orthonormal and span the same subspace. To see this, let Q be $k \times k$ orthogonal matrix; then the columns of the matrix

$$W_k = V_k\, Q \tag{53}$$

form a second set of orthonormal vectors spanning $\mathcal{R}(V_k)$, and $W_k\, W_k^T = V_k Q\, Q^T V_k^T = V_k V_k^T$. Another way to state this is to observe that $V_k V_k^T$ is an orthogonal projection matrix.

This fact allows us – for each estimation algorithm – to choose a new set of vectors $w_1, \ldots, w_n$ that may better describe the estimate $\hat{s}'$ than the vectors $v_1, \ldots, v_n$, knowing that $\hat{s}'$ stays the same. And since these vectors define FIR filter coefficients in a filter interpretation, this means that we are free to choose filters as long as (53) is satisfied.

In particular, if we want to compare the output of two rank-reduction algorithms, then we can try to choose the vectors $w_1, \ldots, w_n$ for the two algorithms such that these vectors are as similar as possible. The more similar the filters, the more similar the estimates.

The solution to the problem of choosing these vectors comes in the form of the *canonical vectors* associated with the subspaces spanned by the columns of the V_k-matrices for the two algorithms in consideration.

To illustrate this, let us compare the truncated SVD and ULV algorithms which produce LS estimates. We work with the two matrices V_k and $V_{L,k}$, and we let $\mathcal{V}_k = \mathcal{R}(V_k)$ and $\mathcal{V}_{L,k} = \mathcal{R}(V_{L,k})$ denote the subspaces spanned by the columns of these two matrices. If

$$V_k^T V_{L,k} = U_\Theta\, \Theta\, V_\Theta^T \tag{54}$$

is the SVD of the cross-product matrix, then the *canonical vectors* w_i and $w_{L,i}$ are the columns of

$$W_k = V_k U_\Theta \qquad \text{and} \qquad W_{L,k} = V_{L,k} V_\Theta. \tag{55}$$

The singular values appearing in Θ are termed the *canonical correlations*, and they are equal to the cosines of the canonical angles $\theta_1, \ldots, \theta_k$. I.e.,

$$\Theta = \mathrm{diag}\,(\cos(\theta_1), \ldots, \cos(\theta_k)) \tag{56}$$

with $0 \le \theta_1 \le \theta_2 \le \cdots \le \theta_k$. See [2], [3] and [16] for more details.

We emphasize the following geometric interpretation of the canonical angles and vectors. The smallest canonical angle θ_1 is the smallest angle between any two vectors v and v_L in $\mathcal{V}_k$ and $\mathcal{V}_{L,k}$, respectively, and it attained for $v = w_1$ and $v_L = w_{L,1}$. The second canonical angle θ_2 is the smallest angle between any two vectors v and v_L orthogonal to w_1 and $w_{L,1}$ in $\mathcal{V}_k$ and $\mathcal{V}_{L,k}$, and it is attained for $v = w_2$ and $v_L = w_{L,2}$; etc.

Hence, canonical vectors associated with small canonical angles define subspaces of $\mathcal{V}_k$ and $\mathcal{V}_{L,k}$ that are as close as possible, and zero canonical angles define canonical vectors in the intersection of the subspaces $\mathcal{V}_k$ and $\mathcal{V}_{L,k}$. Zero canonical vectors are always present when k is greater than $n/2$, because $\mathcal{V}_k$ and $\mathcal{V}_{L,k}$ (being subspaces of $\mathbb{R}^n$) must have a nontrivial intersection of dimension $2k - n$:

$$\theta_1 = \cdots = \theta_{2k-n} = 0 \quad \text{when} \quad 2k > n.$$

We can now compare the truncated SVD and ULV algorithms by comparing the *canonical FIR filters* determined by the canonical vectors $w_1, \ldots, w_k$ and $w_{L,1}, \ldots, w_{L,k}$. If $k > n/2$ then we are sure that $2k - n$ of these filters are identical, and if some of the nonzero canonical angles θ_i are small then the associated filters are also guaranteed to be similar.

Thus, small (and zero) canonical angles define FIR filters for the two algorithms that extract very similar signal components.

Of course, there is more to this analysis than merely the canonical angles. Even if θ_i is not very small, meaning that the vectors w_i and $w_{L,i}$ are somewhat different, say, in the 2-norm, the associated filters may have similar properties in the frequency domain. For example, w_i and $w_{L,i}$ may represent bandpass filters with approximately the same center frequency and bandwidth.

Hence, it is the size of the canonical angles θ_i together with the frequency responses of the canonical FIR filters represented by w_i and $w_{L,i}$ that provides a convenient tool for comparison of the similarities and differences in the output signals from the two algorithms characterized by V_k and $V_{L,k}$.

We note that, as pointed out in [3], the most accurate way to compute small canonical angles θ_i is via the singular values of the matrix $(I - V_k V_k^T) V_{L,k}$:

```
for i=1:k
   VLk = VLk - Vk(:,i)*(Vk(:,i)'*VLk);
end
theta = asin(min(1,svd(VLk)));
theta = flipud(theta);
```

10.2. Example

We illustrate the above comparison of the truncated SVD and ULV algorithms with a numerical example using the same data as in §3 and with truncation parameter $k = 12$.

In order to demonstrate the power of our analysis, we use the high-rank algorithm `hulv` to compute the ULV decomposition. This algorithm seeks to compute good approximations to the singular vectors corresponding to the smallest singular values, but we cannot expect

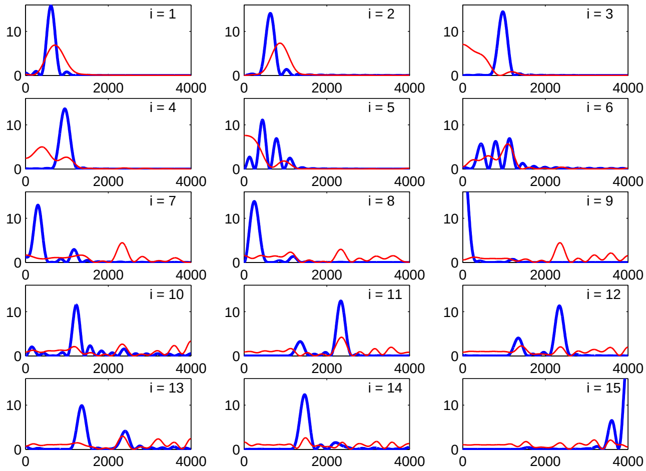

Figure 11. Frequency responses of the FIR filters for the truncated SVD and high-rank ULV algorithms (which produce LS estimates), applied to the test problem in Fig. 5. Thick lines are SVD filters; thin lines are ULV filters.

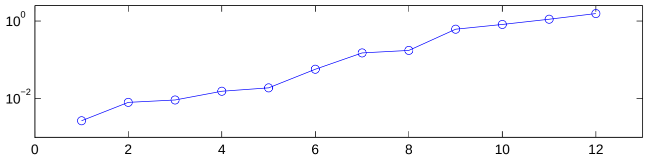

Figure 12. The canonical angles $\theta_1, \ldots, \theta_k$ (in radians) associated with the matrices V_k and $V_{L,k}$ from the SVD and the ULV decomposition computed by the high-rank `hulv` algorithm.

that the principal singular vectors are approximated so well in this algorithm. Hence we cannot expect the FIR filters for the SVD- and ULV-based methods to be very similar, and Fig. 11 confirms this.

Nevertheless the truncated SVD and ULV algorithms produce estimated signals that sound qualitatively the same, in spite of the fact that the FIR filters appear to be quite different. The canonical angles and filters provide an explanation for this. Figure 12 shows the canonical angles $\theta_1, \ldots, \theta_k$ associated with the matrices V_k and $V_{L,k}$, and we see that many of these angles are quite small. Hence we can expect that the two algorithms produce estimates $\hat{s}$ that have very similar signal components lying in a similar *subspaces* of the 12-dimensional signal subspaces $\mathcal{V}_k$ and $\mathcal{V}_{L,k}$ used here.

This is confirmed by the plots in Fig. 13 of the SVD/ULV canonical filters defined by the columns of W_k and $W_{L,k}$ in (55). We see that actually the first 8 canonical filters are very similar. We conclude that for this

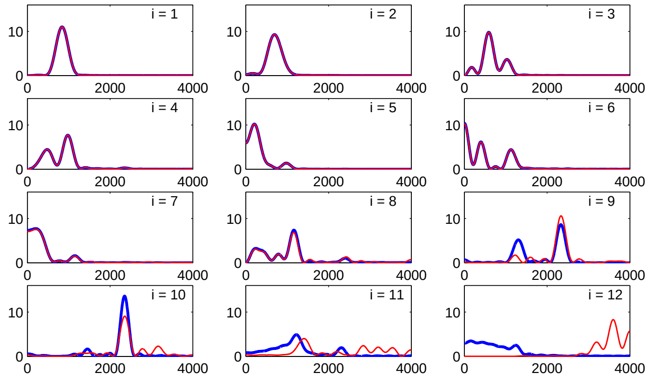

Figure 13. The frequency responses for the SVD/ULV canonical filters defined by the columns of W_k and $W_{L,k}$ in (55). Thick lines are SVD canonical filters; thin lines are ULV canonical filters.

particular noisy signal, the SVD and ULV algorithms produce filtered signals that have very similar signal components, each lying in an 8-dimensional subspace of the respective signal subspaces for the two methods. This explains why the two estimates sound so similar, despite the fact that the columns of V_k and $V_{L,k}$ are quite different.

11. CONCLUSION

In this paper we surveyed the definitions and use of diagonal matrix decompositions (eigenvalue and singular value decompositions) and rank-revealing matrix decompositions (ULV, URV, VSV, ULLV and ULLIV) in single-channel subspace-based noise reduction algorithms for speech signals, and we illustrated the algorithms with working Matlab code and speech enhancement examples. We have further provided finite-duration impulse response (FIR) filter representations of the noise reduction algorithms and derived closed-form expressions for the FIR filter coefficients. Moreover, we have introduced a new analysis tool called canonical filters which allows us to compare the behavior and performance of the subspace based algorithms in the frequency domain.

ACKNOWLEDGEMENT

We thank Nicola Mastronardi for giving us the opportunity to present some of this material at the international workshop in Bari, 2006, where we received constructive feedback from many participants.

APPENDIX: MV and TDC ESTIMATES

The following derivation of the SVD-based MV estimate was given in [7].

Lemma A.1. Let the SVDs of $\overline{H}$ and H be given by (12) and (11), and let E satisfy (13). Then the two SVDs are related by

$$\begin{aligned}
(U_1, U_2) &= \Big((\overline{U}_1\overline{\Sigma}_1 + EV_1)\Sigma_1^{-1}, (m\eta^2)^{-1/2}EV_2\Big) \\
\begin{pmatrix}\Sigma_1 & 0\\ 0 & \Sigma_2\end{pmatrix} &= \begin{pmatrix}(\overline{\Sigma}_1^2 + m\eta^2 I_k)^{1/2} & 0 \\ 0 & m\eta^2 I_{n-k}\end{pmatrix} \\
(V_1, V_2) &= (\overline{V}_1, \overline{V}_2).
\end{aligned}$$

Proof. Inserting the SVD of $\overline{H}$ into $H = \overline{H} + E$ and using that $E = E\,\overline{V}_1\overline{V}_1^T + E\,\overline{V}_2\overline{V}_2^T$, we get $H = Z\,\overline{V}^T$ with

$$Z = (Z_1, Z_2) = \big(\overline{U}_1\overline{\Sigma}_1 + E\overline{V}_1, E\overline{V}_2\big).$$

We have

$$\begin{aligned}
Z_1^T Z_1 &= \overline{\Sigma}_1\overline{U}_1^T\overline{U}_1\overline{\Sigma}_1 + \overline{\Sigma}_1\overline{U}_1^T E\overline{V}_1 \;+ \\
&\qquad \overline{V}_1^T E^T\overline{U}_1\overline{\Sigma}_1 + \overline{V}_1^T E^T E\overline{V}_1 \\
&= \overline{\Sigma}_1^2 + m\eta^2 I_k \\
Z_2^T Z_2 &= \overline{V}_2^T E^T E\overline{V}_2 = m\eta^2 I_{n-k} \\
Z_2^T Z_1 &= \overline{V}_2^T E^T\overline{U}_1\overline{\Sigma}_1 + \overline{V}_2^T E^T E\overline{V}_1 = 0
\end{aligned}$$

and thus we can write $Z = (Z\,D^{-1})D$ with

$$D = \begin{pmatrix}(\overline{\Sigma}_1^2 + m\eta^2 I_k)^{1/2} & 0 \\ 0 & (m\eta^2)^{1/2} I_{n-k}\end{pmatrix}.$$

By comparing the SVDs of H and $\overline{H}$ it follows that $U = Z\,D^{-1}$, $\Sigma = D$, and $V = \overline{V}$. □

As a consequence of this Lemma we have

$$U_1^T\overline{U}_1 = \Sigma_1^{-1}\big(\overline{\Sigma}_1\overline{U}_1^T\overline{U}_1 + V_1^T E^T\overline{U}_1\big) = \Sigma_1^{-1}\overline{\Sigma}_1$$

$$U_1^T\overline{U}_2 = \Sigma_1^{-1}\big(\overline{\Sigma}_1\overline{U}_1^T\overline{U}_2 + V_1^T E^T\overline{U}_2\big) = 0$$

$$U_2^T\overline{U} = (m\eta^2)^{-1/2}V_2^T E^T\overline{U} = 0$$

and thus

$$U^T\overline{U} = \begin{pmatrix}\Sigma_1^{-1}\overline{\Sigma}_1 & 0 \\ 0 & 0\end{pmatrix}.$$

The matrix $W_{\rm mv}$ that solves (19) is $W_{\rm mv} = H^\dagger\overline{H}$, where $H^\dagger = V\,\Sigma^{-1}U^T$ is the pseudoinverse of H, and thus

$$\widehat{H}_{\rm mv} = H\,H^\dagger\overline{H} = U\,U^T\,\overline{U}\,\overline{\Sigma}\,\overline{V}^T = U_1\,\Sigma_1^{-1}\overline{\Sigma}_1^2 V^T.$$

Using the relation $\overline{\Sigma}_1^2 = \Sigma_1^2 - m\eta^2 I_k$, we immediately obtain (20).

The derivation of the UTV- and VSV-based MV estimate is new; it follows that of the SVD. Note that *we must assume that the off-diagonal blocks in the UTV and VSV decompositions are zero* (in practice, the norm of the off-diagonal block is small). Hence, in our derivation the UTV and VSV decompositions take the block diagonal form

$$\begin{aligned}
\overline{H} &= \overline{U}_T\begin{pmatrix}\overline{T}_{11} & 0 \\ 0 & 0\end{pmatrix}\overline{V}_T^T \\
H &= U_T\begin{pmatrix}T_{11} & 0 \\ 0 & T_{22}\end{pmatrix}V_T^T,
\end{aligned}$$

where T denotes either L, R or S.

Lemma A.2. Assuming that E satisfies (13), the two above decompositions satisfy

$$\begin{aligned}
U_T &= \Big((\overline{U}_{T1}\overline{T}_{11} + EV_{T1})T_{11}^{-1}, (m\eta^2)^{-1/2}EV_{T2}\Big) \\
T &= \begin{pmatrix}\mathtt{chol}(\overline{T}_{11}^T\overline{T}_{11} + m\eta^2 I_k) & 0 \\ 0 & (m\eta^2)^{1/2} I_{n-k}\end{pmatrix} \\
V_T &= \overline{V}_T,
\end{aligned}$$

where $\mathtt{chol}(\cdot)$ denotes the Cholesky factor.

Proof. We insert the decomposition of $\overline{H}$ into $H = \overline{H} + E$ and use $E = E\overline{V}_{T1}\overline{V}_{T1}^T + E\overline{V}_{T2}\overline{V}_{T2}^T$ to obtain $H = Z\,\overline{V}_T^T$ with

$$Z = (Z_1, Z_2) = \big(\overline{U}_{T1}\overline{T}_{11} + E\overline{V}_{T1}, E\overline{V}_{T2}\big).$$

We have

$$\begin{aligned}
Z_1^T Z_1 &= \overline{T}_{11}^T\overline{U}_{T1}^T\overline{U}_{T1}\overline{T}_{11} + \overline{T}_{11}\overline{U}_{T1}^T E\overline{V}_{T1} \;+ \\
&\qquad \overline{V}_{T1}^T E^T\overline{U}_{T1}\overline{T}_{11} + \overline{V}_{T1}^T E^T E\overline{V}_{T1} \\
&= \overline{T}_{11}^T\overline{T}_{11} + m\eta^2 I_k \\
Z_2^T Z_2 &= \overline{V}_{T2}^T E^T E\overline{V}_{T2} = m\eta^2 I_{n-k} \\
Z_2^T Z_1 &= \overline{V}_{T2}^T E^T\overline{U}_{T1}\overline{T}_{11} + \overline{V}_{T2}^T E^T E\overline{V}_{T1} = 0
\end{aligned}$$

and thus we can write $Z = (Z\,D^{-1})D$ with

$$D = \begin{pmatrix}\mathtt{chol}(\overline{T}_{11}^T\overline{T}_{11} + m\eta^2 I_k) & 0 \\ 0 & (m\eta^2)^{1/2} I_{n-k}\end{pmatrix}$$

By comparing the decompositions of H and $\overline{H}$ it follows that $U_T = Z\,D^{-1}$, $T = D$, and $V_T = \overline{V}_T$. □

As a consequence of this lemma we have

$$U_T^T\overline{U}_T = \begin{pmatrix}T_{11}^{-T}\overline{T}_{11}^T & 0 \\ 0 & 0\end{pmatrix}$$

(the derivation is similar to that for the SVD). Hence the UTV-based estimate is

$$\begin{aligned}
\widehat{H}_{\rm mv} &= H\,H^\dagger\overline{H} \;=\; U_T U_T^T\overline{U}_T\overline{T}\,\overline{V}^T \\
&= U_{T1}T_{11}^{-T}\,\overline{T}_{11}^T\overline{T}_{11}V_{T1}^T \\
&= U_{T1}T_{11}\,T_{11}^{-1}T_{11}^{-T}(T_{11}^T T_{11} - m\eta^2 I_k)V_{T1}^T \\
&= U_{T1}T_{11}\,(I_k - m\eta^2 T_{11}^{-1}T_{11}^{-T})V_{T1}^T.
\end{aligned}$$

This is the result given in Table 2.

We now turn to the SVD-based TDC estimate, and we follow the derivation from [12]. The Lagrange function for the constrained problem in (21) is

$$L(W,\lambda) = \|\overline{H}\,W - \overline{W}\|_F^2 + \tilde{\lambda}\,(\|W\|_F^2 - m\alpha^2)$$

where $\tilde{\lambda}$ is the Lagrange parameter for the constraint. Differentiation with respect to the elements in W yields

$$L' = 2\overline{H}^T(\overline{H}\,W - \overline{H}) + 2\tilde{\lambda}W,$$

and setting $L' = 0$ we obtain the condition

$$(\overline{H}^T\overline{H} + \tilde{\lambda}\,I)\,W = \overline{H}^T\overline{H}.$$

Thus

$$\begin{aligned} W_{\mathrm{tdc}} &= \left(\overline{H}^T\overline{H} + \tilde{\lambda}I\right)^{-1}\overline{H}^T\overline{H} \\ &= \overline{V}\left(\overline{\Sigma}^2 + \tilde{\lambda}I_k\right)^{-1}\overline{\Sigma}^2\,\overline{V}^T \\ &= \overline{V}_1\left(\overline{\Sigma}_1^2 + \tilde{\lambda}I_k\right)^{-1}\overline{\Sigma}_1^2\,\overline{V}_1^T. \end{aligned}$$

When we set $\tilde{\lambda} = \lambda\,m\eta^2$, multiply with H, and insert $\overline{\Sigma}_1^2 = \Sigma_1^2 - m\eta^2 I_k$ and $\overline{V}_1 = V_1$ from Lemma A.1, then we obtain (22).

The UTV- and VSV-based TDC estimates are derived analogously, using again the above block-diagonal decompositions:

$$\begin{aligned} W_{\mathrm{tdc}} &= \overline{V}_T\left(\overline{T}^T\overline{T} + \lambda\,m\eta^2 I_k\right)^{-1}\overline{T}^T\overline{T}\,\overline{V}_T^T \\ &= \overline{V}_{T1}\left(\overline{T}_{11}^T\overline{T}_{11} + \lambda\,m\eta^2 I_k\right)^{-1}\overline{T}_{11}^T\overline{T}_{11}\,\overline{V}_{T1}^T. \end{aligned}$$

Multiplying with H and inserting $\overline{T}_{11}^T\overline{T}_{11} = T_{11}^T T_{11} - m\eta^2 I_k$ and $\overline{V}_{T1} = V_{T1}$ we obtain the result in Table 2.

REFERENCES

1. G. Adams, M. F. Griffin and G. W. Stewart, "Direction -of-arrival estimation using the rank-revealing URV decomposition;" in *1991 Int. Conf. Acoust., Speech, Signal Proc.*, Toronto, Ont., April 1991.
2. Å. Björck, *Numerical Methods for Least Squares Problems*, Philadelphia, PA: SIAM, 1996.
3. Å. Björck and G. H. Golub, "Numeerical methods for computing angles between linear subspaces," *Math. Comp.*, vol. 27, 579–594, 1973.
4. A. W. Bojanczyk and J. M. Lebak, "Downdating a ULLV decomposition of two matrices;" in J. G. Lewis (Ed.), *Proceedings of the Fifth SIAM Conference on Applied Linear Algebra*, Philadelphia, PA: SIAM, 1994; pp. 261–265.
5. J. R. Bunch and C. P. Nielsen, "Updating the singular value decomposition," Numer. Math., vol. 31, pp. 111–129, 1978.
6. J. A. Cadzow, "Signal enhancement – a composite property mapping algorithm," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. 36, no. 1, pp. 49–62, 1988.
7. B. De Moor, "The singular value decomposition and long and short spaces of noisy matrices," *IEEE Trans. Signal Processing*, vol. 41, no. 9, pp. 2826–2838, 1993.
8. B. De Moor, "Total least squares for affinely structured matrices and the noisy realization problem," *IEEE Trans. Signal Processing*, vol. 42, no. 11, pp. 3104–3113, 1994.
9. M. Dendrinos, S. Bakamidis and G. Carayannis, "Speech enhancement from noise: a regenerative approach," *Speech Commun.*, vol. 10, no. 2, pp. 45–57, 1991.
10. S. Doclo and M. Moonen, "GSVD-based optimal filtering for single and multimicrophone speech enhancement," *IEEE Trans. Signal Processing*, vol. 50, no. 9, pp. 2230–2244, 2002.
11. I. Dologlou and G. Carayannis, "Physical representation of signal reconstruction from reduced rank matrices," *IEEE Trans. Signal Processing*, vol. 39, pp. 1682–1684, 1991.
12. Y. Ephraim and H. L. Van Trees, "A signal subspace approach for speech enhancement," *IEEE Trans. Speech Audio Processing*, vol. 3, no. 4, pp. 251–266, 1995.
13. R. D. Fierro, P. C. Hansen and P. S. K. Hansen, "UTV Tools: Matlab templates for rank-revealing UTV decompositions," *Numer. Algo.*, vol. 20, pp. 165–194, 1999.
14. R. D. Fierro and P. C. Hansen, "UTV Expansion Pack: Special-purpose rank-revealing algorithms," *Numer. Algo.*, vol. 40, pp. 47–66, 2005.
15. G. H. Golub, "Numerical methods for solving least squares problems," *Numer. Math.*, vol. 7, pp . 206–216, 1965.
16. G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Baltimore, MD: Johns Hopkins Univ. Press, 1996.
17. M. Gu and S. C. Eisenstat, "Downdating the singular value decomposition," SIAM J. Matrix Anal. Appl., vol. 16, no. 3, pp. 793–810, 1995.
18. P. C. Hansen, "The truncated SVD as a method for regularization," *BIT*, vol. 27, pp. 534–553, 1987.
19. P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*, Philadelphia, PA: SIAM, 1998.
20. P. C. Hansen, "Rank-deficient prewhitening with quotient SVD and ULV decompositions," *BIT*, vol. 38, pp. 34–43, 1998.
21. P. C. Hansen and S. H. Jensen, "FIR filter representations of reduced-rank noise reduction," *IEEE Trans. Signal Processing*, vol. 46, pp. 1737–1741, 1998.
22. P. C. Hansen and S. H. Jensen, "Prewhitening for rank-deficient noise in subspace methods for noise reduction," *IEEE Trans. Signal Processing*, vol. 53, pp. 3718–3726, 2005.
23. P. S. K. Hansen, P. C. Hansen, S. D. Hansen and J. Aa. Sørensen, "Noise reduction of speech signals using the rank-revealing ULLV decomposition," in *Signal Processing VIII Theories and Applications*, vol. 2, Proc. EUSIPCO-96, pp. 967–970, 1996.

24. P. C. Hansen and P. Yalamov, "Computing symmetric rank-revealing decompositions via triangular factorization," *SIAM J. Matrix Anal. Appl.*, vol. 23, pp. 443–458, 2001.
25. Y. Hu and P. C. Loizou, "A subspace approach for enhancing speech corruped by colored noise," *IEEE Signal Processing Lett.*, vol. 9, no. 7, pp. 204–206, 2002.
26. Y. Hu and P. C. Loizou, "A generalized subspace approach for enhancing speech corruped by colored noise," *IEEE Trans. Speech Audio Processing*, vol. 11, no. 4, pp. 334–341, 2003.
27. J. Huang and Y. Zhao, "Energy-constrained signal subspace methods for speech enhancement and recognition," *IEEE Signal Processing Lett.*, vol. 4, pp. 283–285, 1997.
28. F. Jabloun and B. Champagne, "Incorporating the human hearing properties in the signal subspace approach for speech enhancement," *IEEE Trans. Speech Audio Processing*, vol. 11, no. 6, pp. 700–708, 2003.
29. J. Jensen, R. C. Hendriks, R. Heusdens and S. H. Jensen, "Smoothed subspace based noise suppression with application to speech enhancement;" in *Proc. 13th European Signal Processing Conference (EUSIPCO-2005)*, Antalya, Turkey, September 2005.
;" in *1991 Int. Conf. Acoust., Speech, Signal Proc.*, Toronto, Ont., April 1991.
30. S. H. Jensen, P. C. Hansen, S. D. Hansen and J. Aa. Sørensen, "Reduction of broad-band noise in speech by truncated QSVD," *IEEE Trans. Audio Speech Processing*, vol. 3, pp. 439–448, 1995.
31. K. Konstantinides and K. Yao, "Statistical analysis of effective singular values in matrix rank determination," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, no. 5, pp. 757–763, 1988.
32. H. Lev-Ari and Y. Ephraim, "Extension of the signal subspace speech enhancement approach to colored noise," *IEEE Signal Processing Lett.*, vol. 10, no. 4, pp. 104–106, 2003.
33. K. J. R. Liu, D. P. O'Leary, G. W. Stewart and Y.-J. J. Wu, "URV ESPRIT for tracking time-varying signals", *IEEE Trans. Signal Processing*, vol. 42, no. 12, pp. 3441-3448, 1994.
34. F. T. Luk and S. Qiao, "A new matrix decomposition for signal processing," *Automatica*, vol. 30, pp. 39–43, 1994.
35. F. T. Luk and S. Qiao, "A symmetric rank-revealing Toeplitz matrix decomposition," *J. VLSI Signal Process.*, vol. 14, pp. 19–28, 1996
36. F. T. Luk and S. Qiao, "An adaptive algorithm for interference cancelling in array processing;" in F. T. Luk (Ed.), *Advanced Signal Processing Algorithms, Architectures, and Implementations VI*, SPIE Proceedings Vol. 2846, pp. 151-161, 1996.
37. R. J. McAulay and T. F. Quateri, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 34, pp. 744-754, 1986.
38. U. Mittal and N. Phamdo, "Signal/noise KLT based approach for enhancing speech degraded by colored noise," *IEEE Trans. Speech Audio Processing*, vol. 8, pp. 159–167, 2000.
39. M. Moonen, P. Van Dooren and J. Vandewalle, "A singular value decomposition updating algorithm for subspace tracking," *SIAM J. Matrix Anal. Appl.*, vol. 13, pp. 1015–1038, 1992.
40. A. Rezayee and S. Gazor, "An adaptive KLT approach for speech enhancement," *IEEE Trans. Speech Audio Processing*, vol. 9, pp. 87–95, 2001.
41. L. L. Scharf, "The SVD and reduced rank signal processing," *Signal Processing*, vol. 25, no. 2, pp. 113–133, 1991.
42. L. L. Scharf and D. W. Tufts, "Rank reduction for modeling stationary signals," *IEEE Acoust., Speech, Signal Processing*, vol. 35, no. 3, pp. 350–355, 1987.
43. G. W. Stewart, "Rank degeneracy," *SIAM J. Sci. Stat. Comput.*, vol. 5, pp. 403–413, 1984.
44. G. W. Stewart, "An updating algorithm for subspace tracking," IEEE Trans. Signal Proc., vol. 40, pp. 1535–1541, 1992.
45. G. W. Stewart, "Updating a rank-revealing ULV decomposition," SIAM J. Matrix Anal. Appl., vol. 14, pp. 494–499, 1993.
46. G. W. Stewart, *Matrix Algorithms. Volume 1: Basic Decompositions*, PA: SIAM Press, 1998.
47. G. W. Stewart, "The decompositional approach to matrix computation," *Computing in Science and Engineering*, vol. 2, pp. 50–59, 2000.
48. A. J. Thorpe and L. L. Scharf, "Data adaptive rank-shaping methods for solving least squares problems," *IEEE Trans. Signal Proc.*, vol. 43, no. 7, pp. 1591–1601, 1995.
49. D. W. Tufts and R. Kumaresan, "Singular value decomposition and improved frequence estimation using linear prediction," *IEEE Trans. Acoust. Speech, Signal Processing*, vol. 30, no. 4, pp. 671–675, 1982.
50. L. Vanhamme, R. D. Fierro, S. Van Huffel and R. de Beer, "Fast removal of residual water in proton spectra," *J. Magnetic Resonance*, vol. 132, pp. 197–203, 1998.
51. L. Vanhamme, T. Sundin, P. Van Hecke, S. Van Huffel, "MR spectroscopic quantitation: a review of time-domain methods," *NMR in Biomedicine*, vol. 14, no. 4, pp. 233–246, 2001.
52. S. Van Huffel, "Enhanced resolution based on minimum variance estimation and exponential data modeling," *Signal Processing*, vol. 33, no. 3, pp. 333–355, 1993.
53. S. Van Huffel and H. Zha, "An efficient total least squares algorithm based on a rank-revealing two-sided orthogonal decomposition," *Numer. Algo.*, vol. 4, pp. 101-133, 1993.
54. C. H. You, S. N. Koh, S. Rahardja, "An invertible frequency eigendomain transformation for masking-based subspace speech enhancement," *IEEE Signal Processing Lett.*, vol. 12, no. 5, pp. 461–464, 2005.
55. W. Zhong, S. Li and H.-M. Tai, "Signal subspace approach for narrowband noise reduction in speech," *IEE Proc.-Vis. Image Signal Process.*, vol. 152, no. 6, pp. 800–805, 2005.