University of Maryland — College Park

Institute for Advanced Computer Studies — TR–91–166
Department of Computer Science — TR–2813

Direction of Arrival and The Rank-Revealing URV Decomposition*

E. C. Boman†
M. F. Griffen†
G. W. Stewart‡

December, 1991

ABSTRACT

In many practical direction-of-arrival (DOA) problems the number of sources and their directions from an antenna array do not remain stationary. Hence a practical DOA algorithm must be able to track changes with a minimal number of snapshots. In this paper we describe DOA algorithms, based on a new decomposition, that are not expensive to compute or difficult to update. The algorithms are compared with algorithms based on the singular value decomposition (SVD).

*This report is available by anonymous ftp from `thales.cs.umd.edu` in the directory `pub/reports`.

†Advanced Digital Systems, United Technologies Research Center, E. Hartford, CT 06108

‡Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742. This work was supported in part by the Air Force Office of Scientific Research under Contract AFOSR-87-0188.

Direction of Arrival and The Rank-Revealing URV Decomposition

E. C. Boman
M. F. Griffen
G. W. Stewart

December, 1991

ABSTRACT

In many practical direction-of-arrival (DOA) problems the number of sources and their directions from an antenna array do not remain stationary. Hence a practical DOA algorithm must be able to track changes with a minimal number of snapshots. In this paper we describe DOA algorithms, based on a new decomposition, that are not expensive to compute or difficult to update. The algorithms are compared with algorithms based on the singular value decomposition (SVD).

1. Introduction

In many modern subspace-based direction-of-arrival algorithms the eigenvalue decomposition or the SVD is used to compute an estimate of a target's direction in the presence of noise. Unfortunately, the SVD is expensive to compute and difficult to update, which severely limits the applicability of the algorithms. One way out of the difficulty is to observe that the algorithms require only an estimate of rank and orthogonal bases for the signal and orthogonal ("noise") subspaces. Any decomposition providing these items can be used in the place of the SVD. In this paper, we introduce a new decomposition — the URV decomposition — that provides what is needed and is cheap to update. Moreover it does not require an initial decomposition nor does it need to be restarted. The purpose of this paper is to compare the resulting algorithms with algorithms based on the SVD.

2. Rank Revealing URV Decompositions

Let X be an $m \times p$ matrix of effective rank k. By effective rank k we mean that the singular values of X satisfy $\sigma_1 \geq \cdots \geq \sigma_k > \sigma_{k+1} \geq \cdots \geq \sigma_p$, where σ_k is large enough to be regarded as "significant" and σ_{k+1} is small enough to be regarded as "insignificant." Then X can be decomposed in the form

$$X = U \begin{pmatrix} R & F \\ 0 & G \end{pmatrix} V^{\mathrm{H}},$$

where

1. R and G are upper triangular,
2. The smallest singular value of R is approximately equal to σ_k,
3. $\sqrt{\|F\|^2 + \|G\|^2} \cong \sqrt{\sigma_{k+1}^2 + \cdots \sigma_p^2}$.

Since this decomposition makes the rank of X evident on inspection, it is called a *rank revealing* URV decomposition [1].

In practice the choice of k will depend on the size of F and G. For now we will consider F and G small if

$$\nu \stackrel{\mathrm{def}}{=} \sqrt{\|F\|^2 + \|G\|^2} \leq \mathrm{tol}, \tag{2.1}$$

where tol is a user supplied tolerance. We will return to the choice of tol later.

3. Updating The URVD

In this section an algorithm is sketched to update a rank revealing URVD of X when a row x^{H} is appended to X; i.e., when X is replaced by

$$\begin{pmatrix} \beta X \\ x^{\mathrm{H}} \end{pmatrix}. \tag{3.1}$$

Here β is a forgetting factor that damps out the effects of the previous data. The updating procedure determines if the rank has either increased, decreased, or remained the same.

The first step is to compute $(y^{\mathrm{H}} z^{\mathrm{H}}) = x^{\mathrm{H}} V$, where y is of dimension k. The problem then becomes one of updating

$$S = \begin{pmatrix} R & F \\ 0 & G \\ y^{\mathrm{H}} & z^{\mathrm{H}} \end{pmatrix}.$$

There are two cases to consider. The first occurs when $\nu_{\rm new} = \sqrt{(\beta\nu)^2 + \|y\|^2} \leq$ tol. In this case S is reduced to triangular form by a sequence of rotations applied from the left. Since $\nu_{\rm new}$ is less than or equal to the prescribed tolerance, the new URVD satisfies $\nu \leq$ tol, and the approximate rank does not increase. However, it is possible for the rank to decrease. Hence the rank is checked, using a condition estimator [2]. If it has decreased, then R is deflated by computing its rank-revealing URV decomposition.

The second case occurs when $\nu_{\rm new} >$ tol and consequently the rank may have increased. Since the increase in rank can be at most one, the matrix is transformed to upper triangular form in such a way that the small values in all but the first column of F and G are preserved. Then k is increased by one to reflect the possible increase in rank, and the rank of the augmented R is checked as described in the last paragraph.

These updating algorithms require $O(p^2)$ time. They can be implemented on a linear array of p processors in such a way that they require only $O(p)$ time. See [1] for details.

4. An Adaptive Tolerance

In practice the small singular values of X will come from noise, and the user must furnish a tolerance to distinguish them from the singular values associated with the signal. The simple model described in [1] seems to provide an effective tolerance.

Suppose that X has the form $X = \hat{X} + E$, where $\hat{X}$ has rank exactly k. We will assume that the errors are roughly of the same size — say ϵ — so that when the forgetting factor is taken into account, the ith row of E has the form $\beta^{n-i}e_i^{\rm H}$, where the components of the e_i are approximately ϵ in size. Let the columns of V_2 ($V = [V_1 V_2]$) form an orthonormal basis for the error space of $\hat{X}$. Then our tolerance should approximate the norm of $XV_2 = EV_2$ (remember $\hat{X}V_2 = 0$). Now the ith row of EV_2 consists of $p-k$ elements of size roughly $\beta^{n-i}\epsilon$. Consequently,

$$\|EV_2\|^2 \cong (p-k)\epsilon^2 \sum_{i=1}^{n} \beta^{2(n-i)} \leq \frac{(p-k)\epsilon^2}{1-\beta^2}.$$

Consequently the tolerance — call it tol — should be chosen so that

$$\text{tol} \geq \sqrt{\frac{p-k}{1-\beta^2}}\,\epsilon. \tag{4.1}$$

Note that tol should be chosen somewhat larger than the right-hand side of (4.1) to account for statistical fluctuations in the elements of E. If the signal-to-noise ratio (SNR) is reasonable, this should not result in an underestimation of the rank.

5. Updating The DOA

The algorithms we will use to update the DOAs are the minimum-norm (min-norm) method [3] and root-MUSIC. The polynomial rooting form of MUSIC and min-norm will be used only to evaluate the statistical performance of the algorithms and not as a fast algorithm. Note that finding the roots of a polynomial will destroy any advantage gained by using the URVD.

In our simulations we use the standard data model of M narrowband sources impinging on a uniform linear array composed of N ($M < N$) identical, equally spaced, sensors. The narrowband signals with known angular frequency ω impinge on the array from directions, θ_1, θ_2, $\ldots$,θ_M. The reader is referred to [4] for additional details on the signal model and signal covariance matrix.

We may obtain the estimate of the orthogonal subspace required for DOA estimation from the covariance matrix or from the data matrix. In the later case, the problem becomes one of computing a DOA estimate from the updated data matrix

$$\hat{X}_{m+1}^{\mathrm{H}} = \begin{pmatrix} \beta X_m^{\mathrm{H}} \\ x_{m+1}^{\mathrm{H}} \end{pmatrix}. \tag{5.1}$$

Updating the orthogonal subspace of (5.1) is one of updating a URVD.

The first step is to update the subspaces of $\hat{X}_m^{\mathrm{H}}$ using a rank-revealing URVD. Since the matrix U is not needed for min-norm or MUSIC algorithms, it is not necessary to update it. After V and k are updated, the direction estimates are then updated using the min–norm and MUSIC algorithms.

6. Simulation Results

The data for the simulation consist of four uncorrelated sources impinging on a 10-element array from $-15°$, $0°$, $10°$, and $20°$. The forgetting factor β was chosen to be 0.79, representing a 10 snapshot effective window, and the tolerance was set to 3.5. For those simulations using the adaptive tolerance the error, or ϵ, was set to 1.0–the noise variance. The narrowband source frequency was set to 0.2 and the element spacing is half a wavelength. Each of the four sources assumed

SNR (dB)	Signal Subspace	Noise Subspace
0	$0.123 \pm .038$	$O(10^{-9})$
6	$0.052 \pm .019$	$O(10^{-9})$
12	$0.023 \pm .009$	$O(10^{-9})$
18	$0.011 \pm .005$	$O(10^{-4})$
24	$0.005 \pm .002$	$O(10^{-5})$
30	$0.002 \pm .001$	$O(10^{-4})$

Table 6.1: Angle (rads) between SVD and URVD Subspaces for a fixed tolerance, tol=3.5.

an uniformly distributed random phase term from $-\pi$ to π. Moreover, white Gaussian noise uncorrelated with the sources was added to the data snapshots. The number of updates used in all of the estimates was 10, i.e., a total of 10 snapshots.

We chose to run two types of experiments, one to check the distance between subspaces and the other to test the accuracy of the angle estimates. For both of these experiments we varied the signal-to-noise ratio from 0 to 30 dB. All of our experiments were conducted with a 500 record ensemble and the results are tabulated as mean and standard deviation.

In our distance measure we established as "truth" the subspaces computed via the SVD from a 200 snapshot covariance matrix. The distance was then computed between the true subspaces and the "converged" subspaces, i.e., the URVD subspaces at the 10th update or snapshot. Thus we compute the distance between the SVD and URVD signal subspaces, and the distance between the SVD and URVD noise subspaces. The average angle (in radians) between subspaces as a function of SNR are tabulated in Tables 1 and 2. Table 1 is the average distance for a fixed tolerance while Table 2 is the average distance for the adaptive tolerance.

These two tables show that the URVD closely approximates the true subspace even near 0 dB. In Table 1 the subspace representing the noise is practically identical. The average distance of the signal subspace is smaller with an adaptive tolerance but its interesting to note that the noise subspace distance isn't as small as for the fixed tolerance case. Nevertheless the URVD closely approximates the true subspaces computed from a more expensive SVD algorithm.

SNR (dB)	Signal Subspace	Noise Subspace
0	$0.073 \pm .040$	$0.020 \pm .032$
6	$0.028 \pm .018$	$0.017 \pm .018$
12	$0.012 \pm .009$	$0.010 \pm .010$
18	$0.006 \pm .005$	$0.005 \pm .005$
24	$0.003 \pm .002$	$0.003 \pm .002$
30	$0.001 \pm .001$	$0.001 \pm .001$

Table 6.2: Angle (rads) between SVD and URVD Subspaces for an adaptive tolerance, $\epsilon = 1.0$.

The next set of experiments establishes the statistical accuracy of the URVD-based DOA algorithms versus those algorithms based on the SVD of an estimated covariance matrix. The estimated covariance matrix is computed from the 10 snapshot vectors originally used to update the URVD subspaces. In other words we are comparing a block approach versus an updating one. As before we use the converged estimate of the URVD subspaces. The subspaces are then used in the root-MUSIC and min-norm algorithms to compute the angle estimates. In addition to comparing the SVD versus URVD-based algorithms we compare the angle estimates obtained using both the signal subspace and the noise subspace, with the exception of root-MUSIC which was obtained from the noise subspace only. In this experiment we shall only show the mean and standard deviation of the source at $10°$, which is representative of the other three sources.

In Table 3 root-MUSIC is used to estimate the target location. As expected both the SVD and URVD-based algorithms have a large standard deviation at 0 dB SNR but it appears that the URVD based algorithm is actually better. When the SNR increases the URVD-based algorithm has a slightly higher standard deviation. Still the estimate of $10°$ is very good. Note that the entries in the table are rounded to two decimal places.

Next we consider the results using the min-norm algorithm. Table 4 represents the results of using the signal subspace version of the min-norm algorithm. Table 5 on the other hand represents the results of using the noise subspace version of the min-norm algorithm. Both tables show that the URVD-based algorithm exhibits some bias in its estimate whereas the bias in the SVD-based algorithm is evident only at lower SNRs. Again the estimate of target angle is very good.

SNR (dB)	SVD	URVD
0	8.52 ± 8.19	9.91 ± 5.96
6	10.13 ± 0.82	10.08 ± 1.56
12	10.01 ± 0.34	9.99 ± 0.48
18	10.00 ± 0.18	10.00 ± 0.25
24	10.00 ± 0.10	10.00 ± 0.14
30	10.00 ± 0.05	10.00 ± 0.06

Table 6.3: Root-MUSIC estimate of $10°$ for a fixed tolerance, tol=3.5.

SNR (dB)	SVD	URVD
0	9.95 ± 2.18	9.13 ± 3.87
6	10.07 ± 1.08	9.94 ± 1.52
12	10.02 ± 0.48	9.99 ± 0.74
18	10.00 ± 0.23	9.98 ± 0.54
24	10.00 ± 0.12	9.99 ± 0.15
30	10.00 ± 0.06	10.00 ± 0.08

Table 6.4: Min-Norm Estimate of $10°$-Signal Subspace.

SNR (dB)	SVD	URVD
0	9.95 ± 2.18	9.16 ± 3.67
6	10.07 ± 1.08	10.00 ± 1.53
12	10.02 ± 0.48	9.99 ± 0.65
18	10.00 ± 0.23	9.99 ± 0.56
24	10.00 ± 0.12	9.98 ± 0.47
30	10.00 ± 0.06	9.98 ± 0.45

Table 6.5: Min-Norm Estimate of $10°$-Noise Subspace.

SNR (dB)	Signal Subspace	Noise Subspace
0	6.94 ± 5.58	5.31 ± 5.20
6	10.04 ± 1.99	9.87 ± 1.99
12	10.02 ± 0.86	9.98 ± 0.80
18	9.99 ± 0.38	9.99 ± 0.39
24	10.00 ± 0.20	9.99 ± 0.20
30	10.00 ± 0.10	10.00 ± 0.10

Table 6.6: URVD Min-Norm Estimate of 10° with adaptive tolerance, $\epsilon = 1.0$.

Finally the last table represents only the URVD-based min-norm algorithm using both signal and noise subspace versions. Table 6 clearly show that the mean and standard deviation for either subspace method are about the same. The average angle at 0 dB, however, exhibits a large bias error unlike the fixed tolerance algorithm. And the standard deviations are larger up to 18 dB SNR. It appears that for small SNR's either $\epsilon = 1.0$ was not a good choice or that the simple model for deriving an adaptive tol may be inappropriate. A statistical model based on the distribution of the noise might be more suitable for lower SNRs.

7. Summary

In this paper we first introduced what is called a rank revealing URV decomposition and then sketched an algorithm to update it. We also showed how to choose the user supplied tolerance tol. Then we showed that updating the DOA from the orthogonal subspace of a data matrix is essentially one of updating a URVD. In our simulations we examined two things, the average distance between subspaces and the average estimate of one of four target directions. The distance between subspaces showed that on average there is little difference between the subspaces computed from the SVD and those converged subspaces computed from the rank revealing URVD. The average angle estimate using the URVD behaved very much like those computed using the block SVD approach. Though there was some bias evident, it was very small.

We should emphasize that it isn't necessary to know the rank for this algorithm, the updating algorithm automatically tracks the dimensions of the sub-

spaces. This is unlike the SVD-based algorithms which require knowledge of the dimensions of the subspaces. Furthermore the updating algorithm does not require an initial decomposition — we can simply start with the zero matrix — nor does it need to be restarted. Through experiments we have found that the URVD-based DOA algorithms are very stable and perform nearly as well as block SVD algorithms when the number of snapshots are few. Clearly then, updating a high resolution DOA estimate using the URVD offers an attractive alternative to an expensive SVD approach.

References

[1] G. W. Stewart, "An Updating Algorithm For Subspace Tracking," *University of Maryland Computer Science Technical Report CS-TR 2494*, 1990.

[2] N. J. Higham, "A Survey of Condition Number Estimation for Triangular Matrices," *Siam Review*, 29, pp. 575–596, 1987.

[3] R. Kumaresan and D.W. Tufts, "Estimating the Angles of Arrival of Multiple Plane Waves," *IEEE Trans. Aerosp. and Elect. Syst.*, AES-19, pp. 134-139, 1983.

[4] S. Haykin, ed., *Array Signal Processing*, Prentice-Hall, Englewood Cliffs, New Jersey, 1984.