Lecture Notes in Control and Information Sciences

Edited by A.V. Balakrishnan and M. Thoma

63

Analysis and Optimization of Systems

Proceedings of the Sixth International Conference on Analysis and Optimization of Systems
Nice, June 19–22, 1984

Part 2

Edited by
A. Bensoussan and J. L. Lions

Springer-Verlag
Berlin Heidelberg New York Tokyo 1984

AN ADAPTIVE SINGULAR VALUE DECOMPOSITION ALGORITHM AND ITS APPLICATION TO ADAPTIVE REALIZATION.

J. Vandewalle, J Staar, B. De Moor, J. Lauwers.
ESAT Laboratory Katholieke Universiteit Leuven
Kard. Mercierlaan 94
B-3030 Heverlee
Belgium
tel 016/220931

Abstract:

In this paper we present an algorithm ASVD for the computation of the singular value decomposition (SVD). The method presented is a power method for calculating the largest triplets of the SVD of a matrix A when multiplication is "cheap". The method used bears a lot of similitude with the power method for finding the eigenvalues of a symmetric matrix M. The triplets are found one after another and also some deflation techniques (orthogonalization) are used. The algorithm can take profit of the SVD of slightly different matrices and it is based on the geometric properties of the SVD. Tests have shown that it is more efficient than Golub's algorithm if only the dominant part of the SVD of a long sequence of slowly varying matrices is needed. Also storage efficiency is obtained whenever the matrices are structured.
The paper describes the basic ASVD algorithm, its numerical properties using the shift mechanism, an acceleration method, a computer implementation and its use in adaptive state space realization of noisy impulse responses. It is expected that the new ASVD algorithm and its many strategies will be useful in the domain of signal processing, system theory and automatic control, where SVD is becoming more and more an important concept.

1. Introduction.

In signal processing, automatic control as well as in system theory, the singular value decomposition (SVD) is used increasingly (1-12) as well as for its conceptual as its numerical qualities.In order to com-

pute the SVD, one always recommends the use of Golub's algorithm (2, 11, 13, 14), which is very efficient for the complete high precision SVD of a single full rank matrix. It is however the experience of the authors (3,4) that this algorithm is computationally too involved for most signal and system applications. In the context of adaptive state space realization for example (4), the SVD of a long sequence of slowly varying large block Hankel matrices of low numerical rank has to be computed. In general there exist many signal and system applications where only a small set of singular values are of interest (e.g. all those larger than a certain noise level), or where the required accuracy level of the singular values is limited (e.g. because the data are rather inaccurate), or where a good estimate for the SVD exists (e.g. from a slightly different matrix). It's the purpose of this paper to present a new algorithm for the SVD which, unlike Golub's algorithm, can benefit from these restrictions. The adaptive singular value decomposition algorithm (ASVD) converges one after another to the singular values and vectors in an iterative matrix multiplication process.

In section 2 we describe the basic algorithm and the convergence theorems. The numerical properties are analyzed in section 3 using the shift-mechanism, which is easy to visualize. This avoids many unnecessary computations in the iteration cycle. When the convergence is too slow, a special acceleration step can be applied (section 4). A strategy to exploit these speed ups is developped, implemented on a computer and evaluated in section 5. The use of this algorithm for adaptive realization is given in section 6. In the last section we present the conclusions and emphasize that the basic algorithm allows for many different strategies which should be selected according to the application at hand. The most general description (3) of this algorithm allows for a parallel data flow or analog computer implementation since each of the singular triplets of a singular value, a left and a right singular vector can be computed simultaneously. Here we will only present the recursive version where the triplets are computed one after another. This allows to obtain more useful convergence properties and is valid for all implementations on Von Neumann machines.

In computing the singular value decomposition of a matrix A, it is useful to keep in mind that the ASVD algorithm essentially follows certain trajectories of the discrete time system $x_{k+1} = A^t.A.x_k$. One should however be careful to distinguish the computation performed by ASVD from all the techniques which compute the SVD of a matrix A from the eigenvalue decomposition of $A^t.A$. It is known that the squaring performed in $A^t.A$ is numerically dangerous. It is stressed that the numerical

qualities of ASVD are sound.

2. The basic ASVD algorithm.

Before presenting the algorithm, first the notion of singular value decomposition is recalled (1-3, 11, 13) and some notations are introduced.

Theorem 1. For any real mxn matrix A of rank r there exists a real factorization

$$A = U \cdot \Sigma \cdot V \qquad (1.a)$$

where U and V are square orthonormal matrices and Σ is a pseudodiagonal mxn matrix

$$\Sigma = \begin{bmatrix} \mathrm{diag}(\sigma_1\ \sigma_2\ .\ .\ \sigma_r) & 0 \\ 0 & 0 \end{bmatrix} \qquad (1.b)$$

with the singular values σ_i

$$\sigma_1 \geq \sigma_2 \geq . \ . \ . \ . \geq \sigma_r \qquad (1.c)$$

The colums of u_i of U (resp. rows v_j of V) are called the left (resp. right) singular vectors. The set (u_i, σ_i, v_i) is called the i-th singular triplet. The spaces $S_U^i = \mathrm{Span}(u_1, \ . \ . \ . \ , u_i) \subset R^m$ (resp. $S_V^j = \mathrm{Span}(v_1, \ . \ . \ . \ , v_j) \subset R^n$) are called the i-th principal left (resp. j-th principal right) singular subspaces. For the uniqueness properties and the energy properties (14) of the SVD we refer to the literature. In order to understand the ASVD algorithm it is crucial to realize that any linear map $A: x \longrightarrow y = A \cdot x$ is the result of 3 linear operations $y = U \cdot (\Sigma \cdot (V \cdot x))$: an orthonormal transformation V, a scaling of the axes Σ and another orthonormal transformation U. Let the vector x be described with respect to the orthonormal basis $v_1, \ . \ . \ ., v_n$ as

$$x = \sum_{i=1}^{n} h_i \cdot v_i \qquad (2.a)$$

then from theorem 1 x is mapped by A into

$$y = A \cdot x = \sum_{i=1}^{r} (\sigma_i . h_i) . u_i \qquad (2.b)$$

In other words, each of the coordinates in the V-bases is scaled with a singular value and reconstructed in the U-bases. Clearly, the components with the lowest index are more amplified than those with the largest. Analogously A^t maps

$$s = \sum_{i=1}^{m} \sigma_i \cdot u_i \qquad (3.a)$$

into

$$t = A^t \cdot s = \sum_{i=1}^{r} (\sigma_i \cdot \sigma_i) \cdot v_i \qquad (3.b)$$

Again the components σ_i with the lowest index are more amplified than those with the largest. The net result is that successive multiplications on the left with A and A^t forces a starting vector towards the singular vector with the greatest singular value. Based on these ideas it is quite natural to describe the basic ASVD algorithm as follows:

Computation of the R dominant triplets (u_i, σ_i, v_i) of matrix A:

For i = 1, 2, . . . , R perform the following process until it has converged:

Step 1: Set n=0, choose a vector $e_i^{(0)}$ as initial guess for u_i.

Step 2: Set n=n+1. Obtain the vector $f_i^{(n)}$ as a result of the multiplication $A^t.e_i^{(n-1)}$, orthogonalization with respect to S_V^{i-1} and normalization.

Step 3: Set n=n+1. Obtain the vector $e_i^{(n)}$ as a result of the multiplication $A.f_i^{(n-1)}$, orthogonalization with respect to S_U^{i-1} and normalization. $s_i^{(n)}$ is the lenght of the vector before normaliz.

Step 4: Verifywhether the changes $e_i^{(n-1)} - e_i^{(n-3)}$, $s_i^{(n)} - s_i^{(n-1)}$, $f_i^{(n)} - f_i^{(n-2)}$ are sufficiently small. If not, return to Step 2. If so the i-th singular triplet is obtained as

$$u_i = e_i^{(n-1)}, \quad \sigma_i = s_i^{(n)}, \quad v_i = f_i^{(n)} \qquad (4)$$

The convergence of this process for sufficiently large n to the correct values (4) is guaranteed for almost all initial values by the following theorem and its corollary under the assumption of infinite precision arithmetic.

Theorem 2. For any mxn matrix with SVD (1) and for the vectors determined in the basic ASVD algorithm, one has:

$$\forall e \in S_U^p : \quad (A^t.e) \in S_V^p \qquad (5.a)$$

$$\forall f \in S_V^p : \quad (A.f) \in S_U^p \qquad (5.b)$$

$$e_p^{(0)} = \sum_{i=p}^{r} h_i.v_i : \qquad e_p^{(n)} = \sum_{i=p}^{r} h_i {}^{n}_{i} v_i \Big/ \sqrt{\sum_{i=p}^{r} h_i^2 {}^{2n}_{i}} \qquad (5.c)$$

$$s_p^{(n)} = \sqrt{\sum_{i=p}^{r} h_i^2 {}^{2n}_{i}} \Big/ \sum_{i=p}^{r} h_i^2 {}^{2n-2}_{i} \qquad (5.d)$$

Proof: use the properties of the SVD and (3)&(4) (3,16,17)

Corollary 1: If $\sigma_p > \sigma_{p+1} > \sigma_{p+2}$ then the basic ASVD algorithm satisfies:

with $e_p^{(0)} = \sum_{i=p}^{r} h_i v_i$, $h_p \neq 0$, $h_{p+1} \neq 0$ and for sufficiently large n :

$$\left\| e_p^{(n)} - v_p \right\| = \left\| \Delta_p^{(n)} \right\| = \frac{h_{p+1}}{h_p} \left(\frac{\sigma_{p+1}}{\sigma_p}\right)^n \qquad (6.a)$$

$$s_p^{(n)} - \sigma_p = \frac{1}{2} \left(\frac{h_{p+1}}{h_p}\right)^2 . \left(1 - \frac{\sigma_{p+1}^2}{\sigma_p^2}\right) . \left(\frac{\sigma_{p+1}}{\sigma_p}\right)^{2n} \qquad (6.b)$$

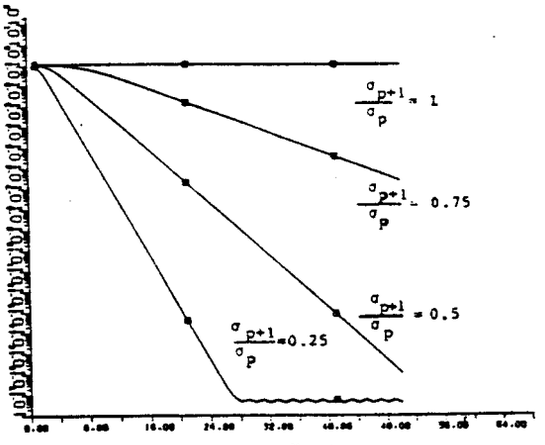


Fig.1. The error $\Delta_n = \left\| e_p^{(n)} - u_p \right\|$ decreases exponentially in terms of the number of iteration cycles. The rate however strongly depends on the ratio σ_{p+1}/σ_p.
For those cases where this ratio approximates 1, the acceleration step of section 4 is needed.

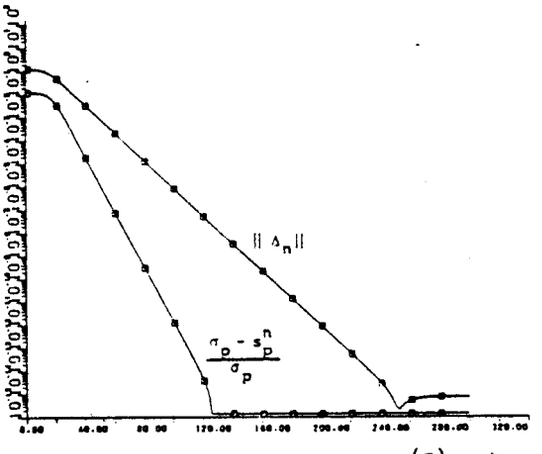


Fig.2. Both the error $\Delta_n = \left\| e_p^{(n)} - u_p \right\|$ on the singular vector u_p and the error on the singular value decrease exponentially but the singular values converge twice as fast as the vectors.

To conclude this section let us remind that the basic ASVD algorithm computes the triplets one after another. Unlike the Golub algorithm one can compute some singular triplets and one can use estimates for the singular values and vectors from nearby matrices and most of all one need not modify the matrix. Hence it is quite useful for sparse or structured (Hankel, Vandermonde, Toeplitz,. . .) matrices.

3. Analysis of the numerical properties using the shift mechanism.

Here we want to analyse the convergence of ASVD with floating point arithmetic (i.e. finite precision). From (2)&(3) and theorem 1, it is clear that the convergence can be easily analyzed if all vectors are represented in terms of the singular basis $u_1, \ldots, u_r$ or $v_1, \ldots, v_r$ and if one combines the two spaces such that $\Phi_i = u_i = v_i$. Though this re-

presentation of the vectors can only be given when the singular vectors are known, also for floating point arithmetic much insight in the mechanism and good strategies are based upon this representation. Any vector $e^{(n)}$ or $f^{(n)}$ can be described as

$$\sum_{i=1}^{r} \mu_i \phi_i \tag{7.a}$$

After a multiplication with A or A^t each component i is scaled by σ_i.

$$\sum_{i=1}^{r} \mu_i \sigma_i \phi_i \tag{7.b}$$

When the coordinates in the ϕ_i-basis are represented on a logarithmic scale one sees that the multiplication causes for each coordinate i a shift by $\log(\sigma_i)$. The normalization then causes a shift of all components until the sum of the coordinates squared is 1. The floating point representation of the vectors (single precision 8 or double 16 decimals) then requires us to look at a window of 8 or 16 units (fig.3).

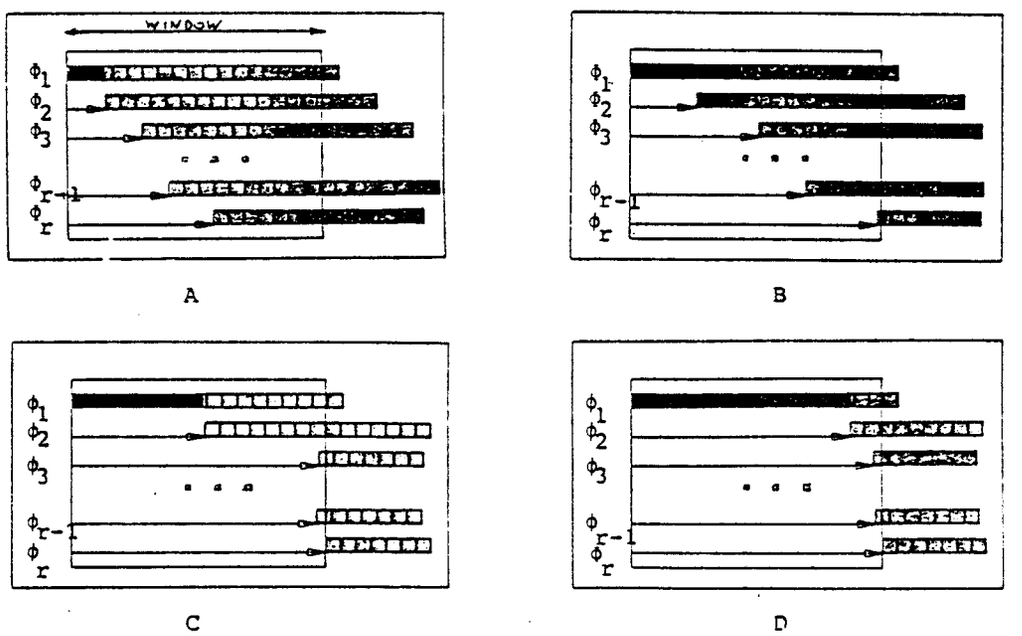


Fig.3. The shift mechanism of ASVD. Each figure describes in logarithmic scale the component of ϕ_i at a certain step in the convergence. As the algorithm proceeds the pattern moves from (A) to (B), (C) and (D). The black bar denotes the precision of the approximation of the singular vector. In each step the accuracy is increased with $\log_{10}(\sigma_1/\sigma_2)$.

Conclusion: The net effect of a multiplication followed by a normalization is that, up to second order effects, the coordinate i is in logarithmic scale shifted by $\log(\sigma_1/\sigma_i)$.

In order to have a practical and efficient algorithm, a number of issues have to be solved.

First, the computation of the p-th triplet can be greatly speeded up by making a good estimate for the p-th triplet during the convergence of the (p-1)-th triplet. The (p-1)-th convergence pattern may provide a good initial guess for the p-th triplet by taking the normalized difference

$$1_p^{(n)} = \frac{e_{p-1}^{(n+2)} - e_{p-1}^{(n)}}{\left\| e_{p-1}^{(n+2)} - e_{p-1}^{(n)} \right\|}$$

It is in general not optimal to take the value at the beginning of the iteration nor at the end (fig.4).

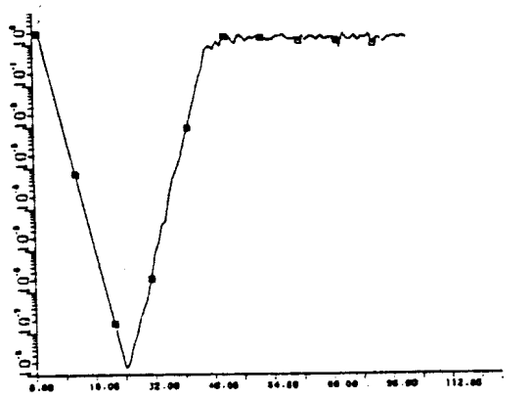
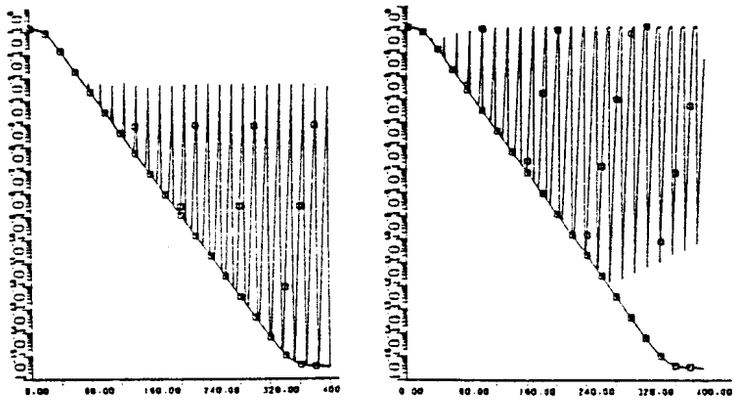


Fig.4. The error $\| 1_p^{(n)} - u_p \|$ of the estimate $1_p^{(n)}$ of the p-th singular vector taken at the n-th step in the iteration of the (p-1)-th triplet. The optimal instant appears when the triplets p+1, p+2, . . have disappeared out of the window of finite precision.

Secondly, since the orthogonalization (Gramm-Schmidt or rather modified Gramm-Schmidt,(16-17)), is time consuming, one may wonder what happens to the convergence pattern of the p-th singular triplet if not during each step an orthogonalization is performed but every t steps (fig.5)



(A) (B)

Fig.5. The error on the p-th singular vector by performing ASVD and orthogonalizing:
(A) every 16 steps.
(B) every 17 steps.

During the t steps between any two orthogonaliztions, a gradual deviation from the standard pattern (fig.1-2) occurs, which is completely eliminated by the orthogonalization. However, if too few orthogonalizations are performed, accuracy is lost (fig.5B). Again, this can be explained and analyzed by the shiftmechanism.

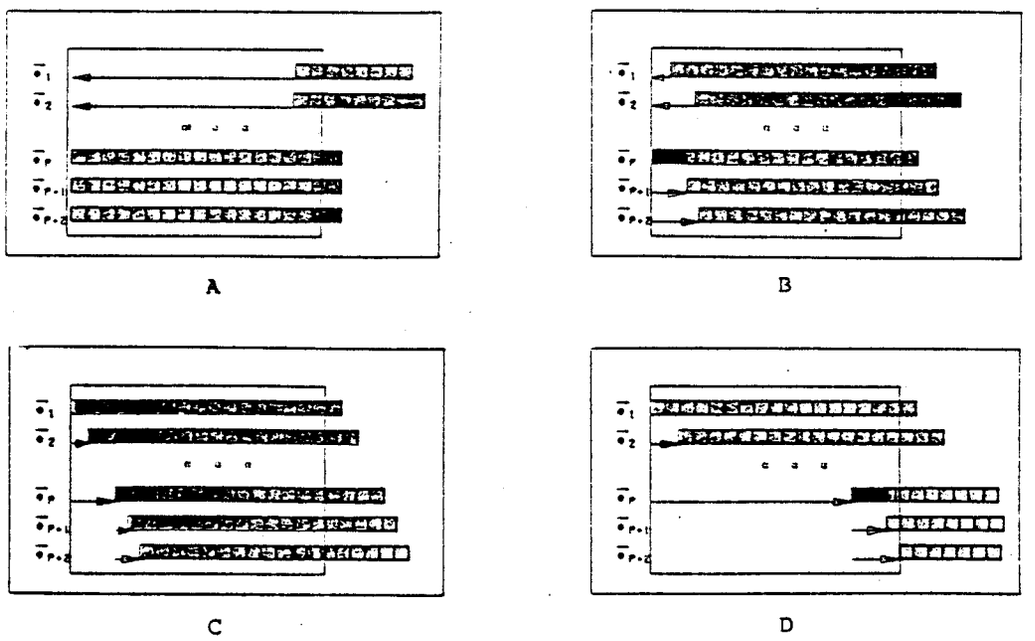


Fig.6. Analysis of the omission of some orthogonalizations in the iteration of the p-th triplet. After orthogonalization (A) the first p-1 contributions are at machine precision. The contributions of the p-1 first triplets grow faster (B&C). If one does not wait with the orthogonalization until the (p+1)-th contribution has disappeared (C) there is no los of accuracy. This condition has allowed to generate a period of t=16 for the example of fig.5. which in that case is the best possible.

4. An acceleration method.

From theorem 1 and fig.1 it is clear that the convergence for the p-th triplet is rather slow when σ_{p+1}/σ_p is close to 1. In this respect it is nice to remember that for $\sigma_{p+1} = \sigma_p$ only the left (resp. right) singular subspaces of the two singular vectors are unique. Then the left (resp. right) singular vectors can be freely chosen as orthogonal vectors in this subspace. Since for σ_{p+1}/σ_p close to 1 the basic ASVD al-

gorithm has more difficulty in separating the p-th triplet from the (p+1)-th, it is more appropriate to iterate together on the p-th and the (p+1)-th triplet. This is the basic idea of the acceleration algorithm. Let ϕ_p and ϕ_{p+1} be the two singular vectors and let e_p and e_{p+1} be two estimates of these singular vectors that are in the same two-dimensional subspace as ϕ_p and ϕ_{p+1}. This subspace is drawn in fig.7. along with the vectors $A^t.e_p$, $A^t.e_{p+1}$, $f_p = A^t.e_p/\|A^t.e_p\|$ and $A.f_p$.

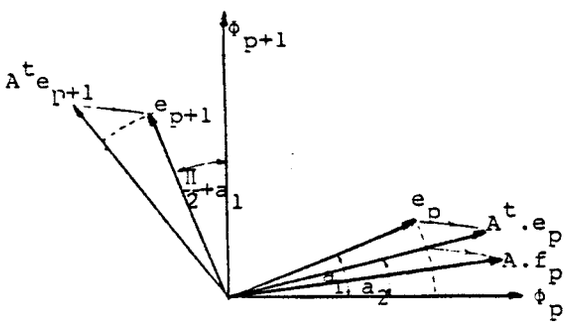


Fig.7. Geometric configuration used in the acceleration method.

Applying simple geometry and the basic properties (1)&(2) of multiplication with A or A^t one can set up (16,17) 4 nonlinear equations which relate the known parameters $s_1 = \|A^t.e_p\|$, $s_2 = \|A^t.e_{p+1}\|$, $t_1 = \|A.f_p\|$ to the unknown parameters σ_p, σ_{p+1} and the angles a_1 and a_2. These nonlinear equations can fortunately be solved analytically (16,17). Hence no further iteration is needed.

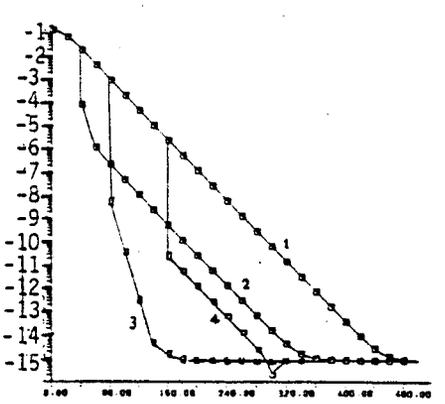


Fig.8. Curve 1 describes the exponential decrease of the error during the iteration of the basic ASVD algorithm. In curves 2, 3 & 4 one acceleration step is performed respectively at n=40, 80 & 160. Clearly, the acceleration has less effect when it is done too soon (curve 2) or too late (curve 4). Also it may happen that the convergence after an acceleratton is faster (curve 3) than before. Based on the shift mechanism this implies that after an acceleration the contribution of the (p+2)-th triplet dominates that of the (p+1)-th.

A detailed analysis has shown that a single acceleration is best performed when the contribution in ϕ_{p+1} after the acceleration with respect to that before is minimal(16). In some implementations it can be worth

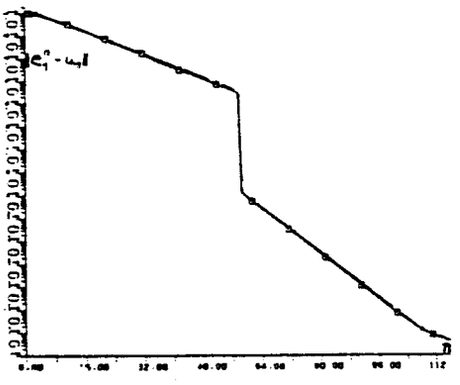
to do an acceleration in each step. Of course one acceleration step requires three matrix multiplications while a regular ASVD step only one.

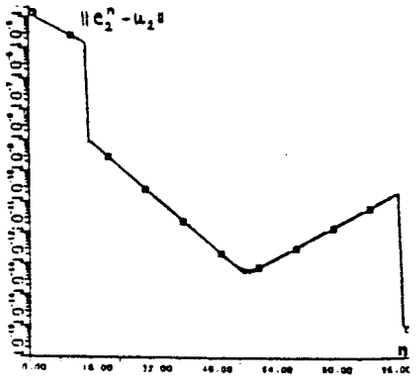
5. Computer implementation of ASVD.

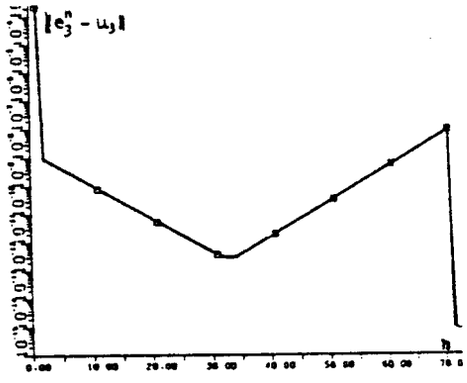
Based on the basic ASVD scheme many different versions of the algorithm with different trade offs between accuracy and efficiency can be generated. In order to dispose of a fully automatic program also techniques for computing σ_p, σ_{p+1}, σ_{p+2} en μ_{p+1}/μ_p in each triplet have been developped. A practical algorithm has been implemented on an IBM 3033 computer and the preliminary tests are promising. A first test example is a 10x10 matrix with singular values: $\sigma_1 = 20.314622$, $\sigma_2 = 17.564908$, $\sigma_3 = 15.298623$, $\sigma_4 = 13.812922$, $\sigma_5 = 11.690173$, $\sigma_6 = 10.182562$, $\sigma_7 = 6.714472$, $\sigma_8 = 2.633963$, $\sigma_9 = 2.382086$, $\sigma_{10} = 2.176375$.
The convergence pattern of some typical triplets are given in fig.9 (triplets 1, 2, 3, 7, 8). Typically the above described version of the ASVD algorithm requires for the computation of 3 singular triplets of a single 10x10 matrix as much time as Golub's algorithm.
Let us now evaluate what happens for a set of slightly different matrices.

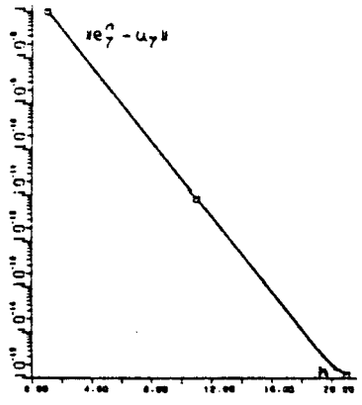
6. Adaptive realization with ASVD.

The state space realization problem aims at finding a state space description for a system which is known by its impulse response. This problem often occurs in system theory and can also be a part of an identification. Algorithms to solve the state space realization problem have already been found in the sixties but have never been worked out in reliable software. Reasons for this are that the numerical qualities of these algorithms are not good (10) and that in practice the impulse responses are corrupted by noise which is also represented (4) in the state space realization. In order to tackle both problems Zeiger and Mc Ewen (8) proposed to find an approximate realization and to use the singular value decomposition. This has later on led to more general algorithms (3, 6, 7, 9) and the formulation of the problem as model reduction. Such algorithms basically perform the singular value decomposition of









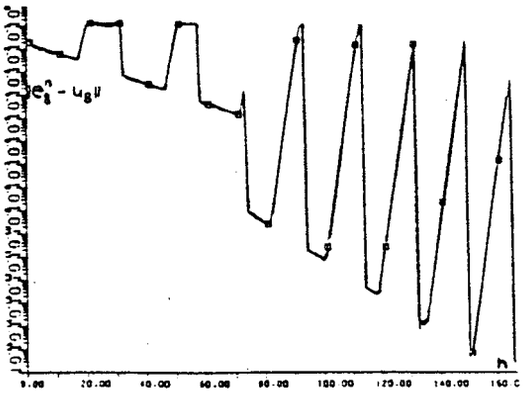


Fig.9 . The convergence $\| e_i^{(n)} - u_i \|$ *for the iteration of triplets i=1, 2, 3, 7, 8. Remark the effect of an acceleration in triplets 1,2 , 3,8 and of an orthogonalization in 2, 3 ,8.*

a block Hankel matrix:

$$\begin{bmatrix} H(1) & H(2) & \cdots & H(M) \\ H(2) & H(3) & \cdots & H(M+1) \\ \vdots & \vdots & \ddots & \vdots \\ H(N) & H(N+1) & \cdots & H(M+N-1) \end{bmatrix}$$

where H(1) . . H(M+N-1) are the lxm sample matrices of the impulse response of a system with m inputs and l outputs. Theoretically, for sufficiently large M and N, the rank of this matrix determines the degree n of the system. However, for noisy impulse responses the Hankelmatrix will be of full rank and hence the degree will be excessive. So, using the SVD (3, 6-9) only the singular triplets, which can be distinguished from the noise, should be considered. In (4) it has also been observed that the number of samples, which is theoretically irrelevant as long as n < M,N dramatically affects the singular values and hence the amount of noise which can be tolerated (several orders of magnitude, fig.10)

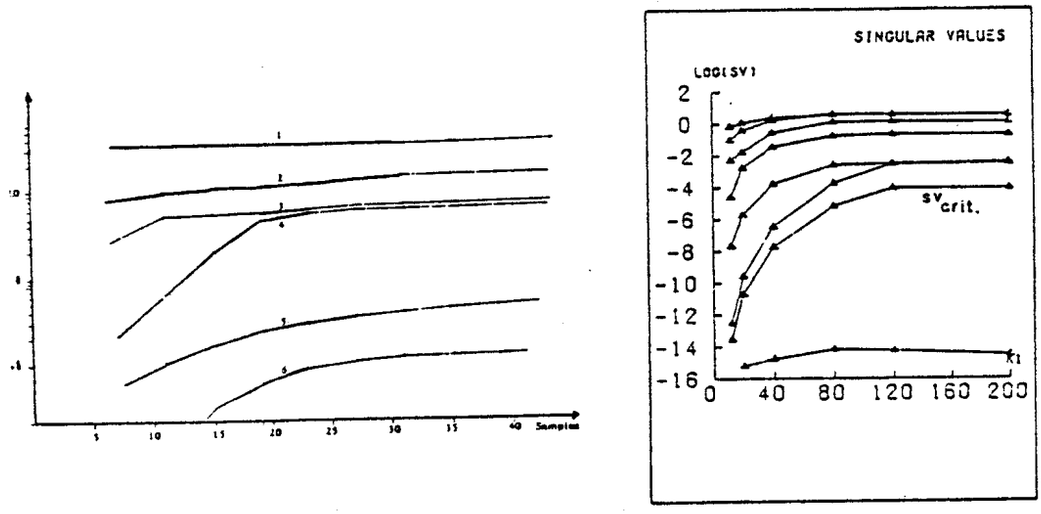


Fig.10. The singular values of two Hankelmatrices in terms of the number of samples of the impulse response. Remark that the last singular value emerges.

Although the experiments with this algorithm were quite promising (4) the on-line implementation with the Golub algorithm is quite time consuming and requires much storage. The ASVD algorithm was derived (3) with the aim of solving both problems. First of all, in an identification context, only few of the singular triplets of the Hankelmatrix have to be calculated. Using the link between energetical concepts and SVD (3) a so called noise criterion can be derived, which determines the number of triplets to be calculated:

$$\sigma_1^2 \geq \sigma_2^2 \geq \ldots \geq \sigma_n^2 > K.R_0.\sigma_v^2 \geq \sigma_{n+1}^2 \cdots$$

where K is a dimension dependent parameter, R_0 is the minimal energy ratio (3) and σ_v = (known noise %) . (max ($\| H(k) \|_F$, k=1,K)).
In this way, the problem of determing the degree n of the system is reduced to a numerical rank problem. The degree is automatically derived from the partial SVD of the involved Hankelmatrix. An example is shown in fig.11.

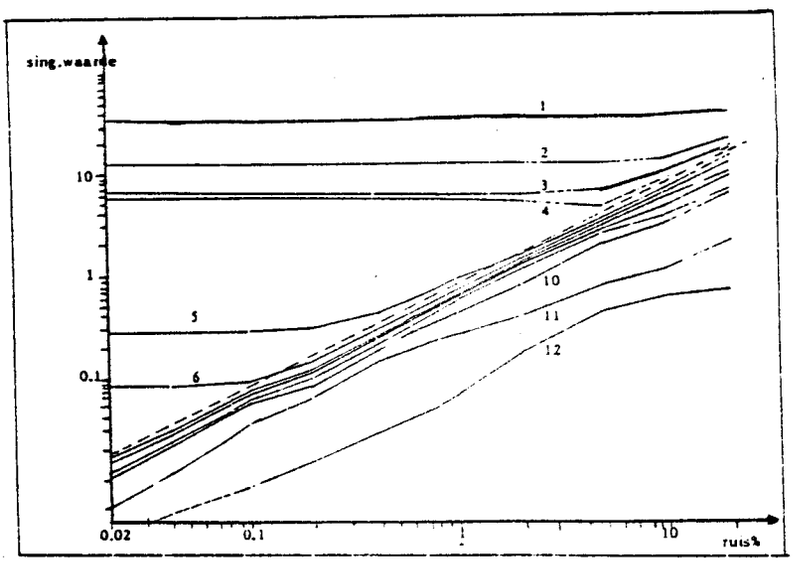


Fig.11. The impulse response of a monovariable 6th order system is stored in a 12x12 Hankelmatrix and poisened by several noise sequencies with increasing energy (standard deviation). The 12 singular values are plotted against the noise %. The noise criterion is plotted in dotted lines with R_0=2. Hence up to 0.1 % of noise the degree is determined correctly to be 6, between 0.1% and 1% the best model degree is 5 and up to 10% the degree is 4. In the neighbourhood of the noise criterion singular values cluster together. In those triplets the use of the acceleration algorithm is unavoidable.

Secondly, for on line identification the SVD of a sequence of slightly varying matrices has to be calculated. Making use of the recursivity of ASVD, one can take the SVD of a previous time step as a starting point for the calculation of the SVD at the present time step.(Fig.12).
Thirdly, the considerable gain in storage of ASVD with respect to the Golub algorithm is based upon the fact that Golub's algorithm operates on the Hankelmatrix entries while ASVD only requires the storage of H(1), ... , H(M+N-1).(Fig.13)
We may conclude that these partial results for adaptive realization with the ASVD algorithm are promising. Time saving properties are obtained by making use of the modular structure and the recursivity and by applying acceleration algorithms. Storage gain is obtained by exploiting the structure of the Hankelmatrices involved.

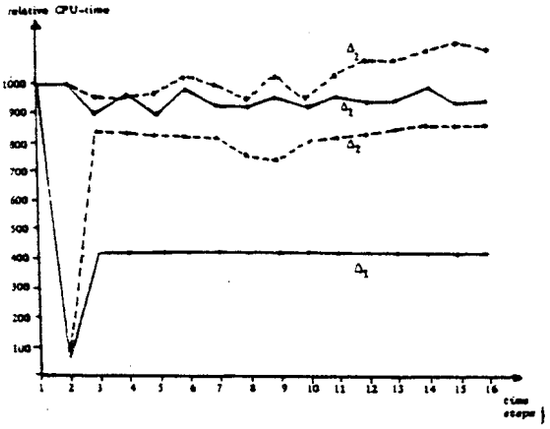


Fig.12. A set of 17 slightly varying 10x12 Hankelmatrices is considered of which the first and the second are equal. The upper curve is the time required for the calculation of 10 triplets of the SVD of each matrix with ASVD when no information of previous time steps is used. The other curves plot the time required for the SVD with ASVD when the starting vectors for timestep T are the singular vectors of timestep T-1. The curves marked with Δ_1 are the results when the impulse response is changing rather slowly. The curves Δ_2 correspond to faster changes. Other speed improvements are still possible since none of the speed refinements of the previous sections are used here in order to show only the time saving by making use of the recursivity.

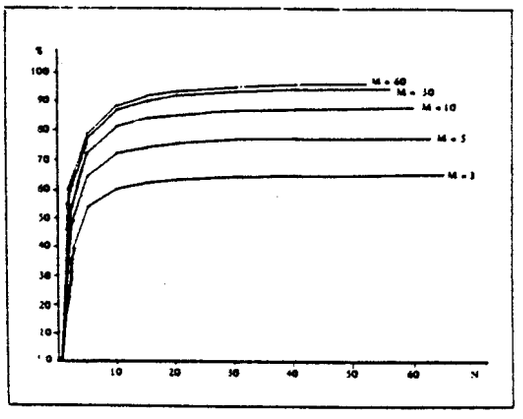


Fig.13. The storage gain obtained by ASVD with respect to the Golub algorithm for MxN matrices.

7. Conclusions.

ASVD is a recursive algorithm which gradually computes the dominant contributions in the SVD of a matrix A. The method bears a lot of similitude with the power method for the symmetric eigenvalue problem (21-22) Deflation techniques, similar to those described in Wilkinson (21) and Parlett (22), as well as acceleration methods are used.

The main advantages of the ASVD algorithm are: (i) Its natural and direct link with a basic property of SVD and hence its transparency for the potential user. (ii) Itssimplicity, the only building blocks being the matrix-vector product, orthogonalization and normalization. (iii) Its complete recursivity. Any earlier approximation cuts down the remaining computation cost.(iv) Its modularity, and hence its flexibility for non-

conventional implementations. (v) The fact that matrix A is not altered during the algorithm, and hence the storage efficiency in the case of highly structured or sparse matrices. (vi) Its numerical reliability (backward stable algorithm) and an easy understanding of the convergence by the shift mechanism.

Its main weak points are: (i) The algorithm is not competitive with Golub's algorithm for a single complete high precision SVD of a full rank matrix on a conventional machine. (ii) Convergence may be very slow for two triplets associated to very close singular values, if appropriate acceleration algorithms are not executed. Such full proof acceleration rules are not yet available for clusters of more than two singular values.

Some tests have already shown that ASVD can be competitive with the Golub algorithm when only the dominant triplets of a sequence of large matrices are needed. Such situations arise in realization and identification, signal processing (12) and control theory: e.g. Quasi-Nyquist loci (20). For the Quasi-Nyquist loci the complex version of this algorithm is feasable and under consideration. Further work will be directed towards the evaluation of other strategies which may be less accurate but faster. In the case of the state space realization this may provide a unification of all recursive algorithms and allow a better trade off between speed and accuracy.

REFERENCES.

1. Special Issue on Multivariable Control Systems, IEEE Trans. Aut. Contr., Vol AC-26, No. 1, 1981.
2. V.C. Klema and A.J. Laub, "The singular Value Decomposition: Its computation and some applications." IEEE Trans. Aut. Contr., Vol AC-25,No 2, p.164-176, 1980.
3. J. Staar, "Concepts for reliable modelling of linear systems with application to on-line identification of multivariable state space descriptions", doct. diss., ESAT laboratory, Kath. Univ. Leuven,June 1982.
4. J. Staar, J. Vandewalle & M. Wemans, "A Comparison of multivariable MBH-algorithms in the presence of multiple poles, and noise disturbing the Markov sequence." Proc. 4-th Int. Conf. on analysis and optimisation of systems (Lecture notes in Control and Information Sciences), p.141-160, Springer Verlag, Versailles, December 1980.
5. J. Staar & J. Vandewalle, " On the numerical implication of the choice of a reference node in the nodal analysis of large circuits", Int. Jrnl. Circuit Theory and Applications, Vol 9, p.488-492, 1981.
6. J. Vandewalle and J. Staar, "On the recursive identification of multivariable state space models", Proc. Symposium on applications of multivariable system theory, Plymouth, October 1982.
7. J. Vandewalle & J. Staar, "Modelling of linear systems: critical examples and numerically reliable approaches", Proc. ISCAS, p.915-918,Rome May 1982.
8. H.P. Zeiger and A.J. Mc. Ewen, "Approximate linear realizations of given dimension via Ho's algorithm", IEEE Trans. Aut. Contr., Vol AC-19 p.153,1974.
9. S.Kung, "A new identification and model reduction algorithm via singular value decomposition", Proc. 12-th Asilomar Conf. Circ. Syst. Nov 1978
10. L.S.De Jong. "Numerical aspects of the recursive realization algorithm", SIAM Journal on Control and optimization, Vol.16, p.646-659,1978.
11. C.L. Lawson and R.J. Hanson, "Solving least squares problems", Prentice Hall Series in Automatic Computation, Englewood Cliffs 1974.
12. J. Vanderschoot, J. Vandewalle, J. Janssen, W. Sansen, H. Van Trappen: "Extraction of weak bioelectrical signals by means of singular value decomposition", sub. to this conference.
13. G.H. Golub and W.Kahan, "Calculating the Singular values and Pseudo-inverse of a matrix", SIAM J. Num. Anal. No.3, p.205-224,1965.
14. G.H.Golub and C.Reinsch, "Singular Value Decomposition and Least squares solutions", Num. Math., Vol.14,p.403-420,1970.
15. J.Staar and J. Vandewalle, "Oriented energy and signal to signal ratios concepts and use" in preparation for sub. to Systems and Control letters.
16. J.Lauwers, "Singuliere waardenontbinding in signaalverwerking; ontwerp en analyse van een efficient algoritme", Master's thesis, ESAT lab., Kath. Univ. Leuven, July 1983.
17. B.De Moor, "Een betrouwbaar adaptief realisatiealgoritme gebaseerd op recursieve singuliere waardenontbinding.", Master's thesis, ESAT Lab., Kath. Univ. Leuven, July 1983.
18. B.Garbow et al., "Matrix eigensystem routines -EISPACK guide extension", (Lecture Notes in Computing Sciences, Vol 51) New York:Springer 1977.
19. J.Dongerra et al.,"Linpack User's guide", Philadelphia, SIAM 1979.
20. Y.S.Hung and A.G.J. Mac Farlane, "Multivariable feedback: A quasi-classical approach", Springer Lecture notes in Control and Information Sciences, Berlin 1982.
21. J.H.Wilkinson, "The algebraic eigenvalue problem",Clarendon Press, Oxford 1965.
22. Beresford N. Parlett, "The symmetric eigenvalue problem." Prentice Hall Series in computational mathemathics, 1980.