MIMO SYSTEMS LOW COMPLEXITY SVD IMPLEMENTATION ANALYSIS

César Benavente Peces(1), Luis Arriero Encinas(1), David Osés del Campo(1),Fco. Javier Ortega González(2), José Manuel Pardo Martín(2).
cesar.benavente@upm.es, larriero@ics.upm.es, doses@ics.upm.es, fjortega@diac.upm.es, jmpardo@diac.upm.es
(1) Dpto. de Ingeniería de Circuitos y Sistemas; Universidad Politécnica de Madrid; E.U.I.T. de Telecomunicación; Ctra. Valencia, km. 7, 28031 Madrid.
(2) Dpto. de Ingeniería Audiovisual y Comunicaciones; Universidad Politécnica de Madrid; E.U.I.T. de Telecomunicación; Ctra. Valencia, km. 7, 28031 Madrid.

***Abstract*- This paper analyses the implementation of the singular value decomposition (SVD) using approximation to the exact computation for MIMO systems in the case of modulation-mode and power assignment set-up. The study developed in the paper focuses on the use of low complexity algorithm with low computational load oriented to the use of devices with limited resources as FPGA, highlighting some of the advantages and drawbacks against more sophisticated devices. The implementation of the SVD is analyzed through the algorithms that efficiently perform the required computations, seeking for computationally efficient solutions that provide parallelism and low complexity. The CORDIC algorithm seems to be a good candidate for this task since it can efficiently compute the singular value decomposition. It is shown that this algorithm provides an efficient tool for SVD computation with appropriate accuracy and the computational complexity obtained and the required resources make it feasible to be implemented on an FPGA device. System performance degradation is analyzed compared with conventional and exact method for SVD obtaining some key conclusions.**

I. INTRODUCCIÓN

MIMO (multiple input multiple output) technology has attracted attention in wireless communications, since it offers significant increases in data transmission rate and link range without the need of providing larger bandwidths or transmitted power and improvement on the bit error rate (BER). The goal of MIMO techniques is achieving higher spectral efficiency (increasing bps/Hz) and reducing the BER. Appropriate data processing and MIMO systems management techniques should be applied to obtain expected benefits from their architecture by establishing different goals in the implementation (e.g. constant data rate, constant power, etc.). Nevertheless, the implementation of MIMO architectures requires some additional complex computations and appropriate system control over conventional communications systems.

Among the algorithms used to provide a proper estimation we find the singular value decomposition. SVD transforms a MIMO channel into multiple single input single output (SISO) channels having unequal gains.

Several solutions are investigated to optimize MIMO system performance and computational load. In order to avoid any signalling overhead, fixed transmission modes are investigated in [1] regardless of the channel quality. The results have shown that not all MIMO layers have to be activated in order to achieve the best BERs.

This paper focuses on the analysis and simulation of the SVD implementation for modulation-mode and power assignment in MIMO systems using low computational complexity algorithms. The cost paid is a drop in the overall system performance as it is shown.

SVD can be implemented in several ways and on different hardware platforms and architectures. DSP (Digital signal processor) or GPP (General Purpose Processor) are powerful devices that provide high processing speed as well as additional resources for computations and some flexibility. FPGA (field-programmable gate array) devices have increased in the last years their capabilities (including speed processing and calculation resources) and functionalities while maintaining the inherent flexibility that allow building appropriate processing architectures. Nevertheless we are interested on simple devices with limited resources and the reduction of computational load.

The benefits are reducing the power consumption and the possibility of using cheaper devices or allowing the allocation of additional functionalities on the same device.

This paper remarks the feasibility of using appropriate approximations to implement the singular value decomposition providing low complexity operations allowing processing time reduction, saving power and required substrate, with a small degradation of the system performance.

The structure of the paper is organized as follows. Section II deals with the description of the system model in order to set the problem to solve. Section III shows the computation of the singular value decomposition describing different possible strategies for its implementation. Section IV introduces the fundamentals of the CORDIC algorithm as a rotation tool and its application to solve the SVD problem through the Jacobi rotations. Section V is aimed to the analysis of the implementation of the SVD using the CORDIC algorithm in FPGA devices, highlighting the advantages and drawbacks. Section VI shows the results obtained from simulation comparing the approximation with the exact approach. Finally, section VII describes the conclusions obtained at the final of this research.

II. MODELLING THE MIMO SYSTEM

In our discussion we consider a single MIMO system with perfect channel state information (PCSI) at both the transmission and reception sides, and no interference from other systems.

The channel is modelled as a flat independent and identically distributed (i.i.d.) Rayleigh channel. Considering a MIMO system with n_T aerials at the transmit side and n_R at the receiver side, the received signal is given by

$$\mathbf{y} = \mathbf{H} \cdot \mathbf{s} + \mathbf{n} \tag{1}$$

where $\mathbf{y}$ is the $n_R \times 1$ received signal vector, $\mathbf{s}$ is the $n_T \times 1$ transmitted signal vector which contains the complex input symbols and $\mathbf{n}$ is the $n_R \times 1$ vector of the additive white Gaussian noise (AWGN) with variance $\mathrm{var}(\mathrm{Re}[n_R]) = \mathrm{var}(\mathrm{Im}[n_R]) = \sigma^2/2$ for both the real and imaginary parts. $\mathbf{H}$ is an $n_R \times n_T$ complex channel gain matrix with entries also having unit magnitude variance. It's assumed that the coefficients of the channel gain matrix $\mathbf{H}$ are i.i.d. Rayleigh distributed with equal variance and that the number of transmit antennas n_T equals the number of receive antennas n_R. Figure 1 shows the block diagram establishing the nature of the problem to be solved. Given the PCSI condition, we can perform a singular value decomposition of the channel matrix $\mathbf{H}$.

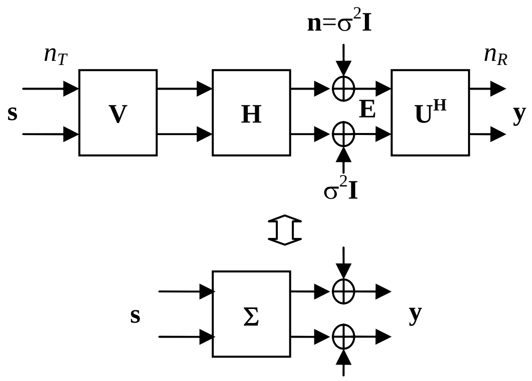

Fig. 1. Block diagram of the problem to be solved.

III. COMPUTING THE SINGULAR VALUE DECOMPOSITION

The SVD of a real $n_R \times n_T$ matrix $\mathbf{H}$ means factorizing into the product of three matrices,

$$\mathbf{H} = \mathbf{U}\Sigma\mathbf{V}^{\mathbf{H}}, \tag{2}$$

where $\mathbf{U}$ and $\mathbf{V}$ are ortogonal $n_R \times n_T$ unitary matrices and $\mathbf{\Sigma} = \mathrm{diag}(\sigma_1,\sigma_2,\ldots,\sigma_n)$ is an $n_R \times n_T$ diagonal matrix containing the singular values of $\mathbf{H}$. The columns of $\mathbf{U}$ and $\mathbf{V}$ are called respectively the left and right singular vectors and σ_i the ith singular value of $\mathbf{H}$. The singular values in $\mathbf{\Sigma}$ may be arranged in any order but they are usually arranged in decreasing order [2].

The singular value decomposition is applied in order to estimate the singular vectors and singular values.

The main source of performance degradation is the deviation of the estimated singular vectors from their true values, which is highly dependent on the accuracy of the channel estimation.

The relationship between the singular value decomposition of $\mathbf{H}$ and the eigenvalue decomposition of $\mathbf{H}^{\mathbf{H}}\mathbf{H}$ is the basis for the algorithms. If the singular values of $\mathbf{H}$ are $\sigma_1,\sigma_2,\ldots,\sigma_n$, then the eigenvalues of $\mathbf{H}^{\mathbf{H}}\mathbf{H}$ are $\sigma_1^2,\sigma_2^2,\ldots,\sigma_n^2$. The right singular vectors, $\mathbf{V}$, are the eigenvectors of $\mathbf{H}^{\mathbf{H}}\mathbf{H}$. The left singular vectors, $\mathbf{U}$, are the eigenvectors of $\mathbf{H}\mathbf{H}^{\mathbf{H}}$.

Several methods can be used to SVD computation. Using serial processing, variants of the QR algorithm (for computing the eigenvalues of a given matrix) can be used to compute the SVD. The basic approach is described in [3]. The drawback with this approach is that the calculation of $\mathbf{H}^{\mathbf{H}}\mathbf{H}$ can lead to a loss of information. A preferable method for computing the SVD is described in [4]. The symmetric QR algorithm is implicitly applied to $\mathbf{H}^{\mathbf{H}}\mathbf{H}$ (in other words, the product $\mathbf{H}^{\mathbf{H}}\mathbf{H}$ is never calculated) in order to simultaneously find $\mathbf{U}$ and $\mathbf{V}$.

The approaches above are very effective on serial processors, but they are not appropriate to parallel implementations. With parallel processing, the preferred methods are variations of the Jacobi algorithm used to compute the eigenvalues of a symmetric matrix. The classical Jacobi algorithm involves pre-multiplication and post-multiplication of $\mathbf{H}$ by proper rotations (Jacobi rotations), with matrices composed of sine and cosine functions.

Parallel implementation implies the storage of two columns of the matrix per processor involved in computations (not necessarily different devices). The determination of those matrices and post-multiplication can be done in parallel with the appropriate architecture. The problem arising with this approach is that the transformation matrices have to move data between processors in order to perform the pre-multiplication that affects the different rows. If processors are independent, the processing delay may be unacceptable. Nevertheless, there are different ways to parallelize processes that can run at the same time on the same device.

Two varieties of algorithms based on Jacobi's method are found: one-sided and two-sided. The last one is computationally more complex than the former and it is not suitable for parallel and pipeline computing. In consequence, to obtain an efficient SVD algorithm the best way is adopting the Hestenes method for one-side transformation [5].

Hestenes' method is based on the classical one-side Jacobi iteration for digitalization of real symmetric matrices. This method is an approach to the SVD computation that avoids successive pre-multiplication of the matrix $\mathbf{H}$. Transformation matrices does not move between processors. To compute the SVD of $\mathbf{H}$, an orthogonal matrix $\mathbf{V}$ is generated such that the transformed matrix $\mathbf{HV} = \mathbf{W}$ has orthogonal columns. Scaling each column of $\mathbf{W}$ so that its 2-norm is unity we get

$$\mathbf{W} = \underline{\mathbf{U}} \cdot \underline{\mathbf{\Sigma}} \tag{3}$$

where $\underline{\mathbf{U}}$ is an $m \times n$ matrix and $\underline{\mathbf{\Sigma}}$ is an $n \times n$ nonnegative diagonal matrix. The corresponding SVD of A is given by

$$\mathbf{H} = \underline{\mathbf{U}} \cdot \underline{\mathbf{\Sigma}} \cdot \underline{\mathbf{V}}^{\mathbf{H}} \tag{4}$$

Comparing equations (4) and (2), we notice the different sizes of $\mathbf{U}$ and $\underline{\mathbf{U}}$ as well as $\mathbf{\Sigma}$ and $\underline{\mathbf{\Sigma}}$. However, since the null columns of $\mathbf{U}$ and $\underline{\mathbf{U}}$ are associated with zero diagonal elements of $\mathbf{\Sigma}$ and $\underline{\mathbf{\Sigma}}$, respectively, the non-null columns of $\mathbf{U}$

and $\underline{\mathbf{U}}$ form orthogonal bases for *range*(**H**) and there is no essential difference between equations (2) and (4). Successive Jacobi rotations are applied to **H** in order to obtain **W**, giving us the following iteration

$$\mathbf{H_{k+1}} = \mathbf{H_k} \cdot \mathbf{J_k} \tag{5}$$

In the case of a 2×2 $n_R=n_T=2$ MIMO systems configuration the rotation matrix becomes

$$\mathbf{J_k} = \begin{pmatrix} \cos\theta_k & -\sin\theta_k \\ \sin\theta_k & \cos\theta_k \end{pmatrix} \tag{6}$$

where θ_k is the rotation angle for the *k* iteration.

Computing the rotation is costly computationally speaking. Furthermore, as this rotation is applied iteratively, the computational load increases in excess. The calculation of trigonometric functions is not an easy task and it claims for proper approaches. This rotation can be performed using efficient algorithms using simple operations with low computational load. The challenge of the practical algorithm is avoiding the computation of trigonometric functions and other ones consuming operations (as products).

IV. Fundamentals of the CORDIC algorithm

The CORDIC algorithm was introduced first to compute trigonometric functions [6] and later generalized to compute linear and hyperbolic functions [7]. It consists of an iterative algorithm which allows computing relatively complex functions by using simple operations, just addition and shift operations. In the most general form a CORDIC algorithm iteration used to perform a rotation can be described by the following set of equations (in a similar way presented in [7]):

$$\begin{aligned} z(k+1) &= z(k) - \sigma \cdot atan(2^{-k}), \\ x(k+1) &= x(k) - y(k)\cdot\sigma\cdot 2^{-k}, \\ y(k+1) &= y(k) + x(k)\cdot\sigma\cdot 2^{-k}, \end{aligned} \tag{7}$$

where *k* ranges from 0 to N, being N the number of iterations used to approximate the rotation angle, and σ takes the value +1 if *z(k)* is equal or greater than 0 and –1 otherwise, and the samples x and y correspond to the coordinates. The target angle equals *z(0)*. Phase micro-rotations are determined by *z* that in the iterative process is forced to become zero.

The range of convergence is $-99.7^\circ \leq z \leq 99.7^\circ$, where 99.7° is the sum of all the possible angles in the list, containing $[-\pi/2, \pi/2]$. This guarantees the maximum $\pi/4$ rotation angles in the Jacobi algorithm.

In [8] the author shows several trigonometric algorithms that can be efficiently implemented using the CORDIC algorithm, including rotation.

In [9] the authors present a specific CORDIC architecture for variable-precision coordinates. This system allows specifying the desired precision to compute the CORDIC rotation, and control the accuracy of the result depending on the application requirements.

V. Implementation analysis

Approximate rotation schemes are in general based on the fact that the annihilated off-diagonal entries are destroyed in later steps of the iterative Jacobi algorithm in such a way that their generation is not strictly justified [10].

For the application of the Jacobi rotations, some trigonometric functions should be computed. The coarsest approximation based on the CORDIC idea is applying just one angle rotating by an angle $\theta_k = arctan(2^{-k})$. This approximation requires a proper discretization of values to become powers of 2, facilitating hardware computations (especially in fixed point arithmetic). The *k* value is bounded by the arithmetic wordlength. Three steps should be performed to implement CORDIC based rotations of the Jacobi algorithm: (1) computing rotation appropriate angle, (2) rotation computation, and (3) scaling.

The basic rotation matrix in the CORDIC based algorithm is described as follows

$$\mathbf{R_k} = K\begin{pmatrix} 1 & -\sigma\cdot 2^{-k} \\ \sigma\cdot 2^{-k} & 1 \end{pmatrix} \tag{8}$$

where σ simply takes the values {+1,-1} and multiplying by a power of two means a bit shift. *K* is the scaling factor correction.

A detailed analysis of the implementation of the CORDIC algorithm for SVD can be found in [11] describing approximations and algorithms complexity.

VI. Results

In order to analyze the behaviour of the MIMO system when using the CORDIC approximation to implement the SVD, the error probability is measured compared with that using exact SVD. The goal is measuring and analyzing the losses due to the use of CORDIC algorithms using as reference the MIMO with exact SVD and a conventional system.

Figure 2 shows the system capacity as a function of the signal-to-noise ratio for different number of transmitting and receiving antennas for exact SVD. When using approximations in the implementation of SVD the channel capacity slightly decreases.

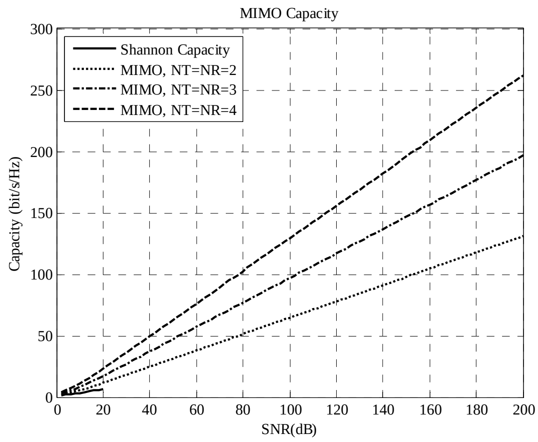

Fig. 2. Capacity as a function of the signal-to-noise ratio for different configurations.

Figure 3 shows the BER of the system as a function of the E_b/N_o ratio. A system with $n_T=4$ transmitting antennas and $n_R=4$ receiving antennas has been considered using a 16-QAM constellation, using both the exact Jacobi rotation in the SVD decomposition and the CORDIC algorithm approximations as described above using fixed point arithmetic with 16 bits word-length and 13 iterations in the CORDIC. The results are compared with from a conventional system. Eventually the results obtained using approximations are worst than those from the MIMO based system. For a BER of 10^{-5} we obtain a loss of approximately 2 dB with respect the exact solution. Nevertheless, the gain respect the conventional system is appreciable. Two different cases have been represented: 8 samples/symbol for the exact and approximate SVD and 4 samples/symbol for approximate SVD. As we decrease the number of samples per bit the performance of the systems degrades.

Figure 4 shows a similar analysis representing the BER as a function of E_b/N_o ratio for 4-QAM transmission. We can compare results obtained for 16-QAM and 4-QAM.

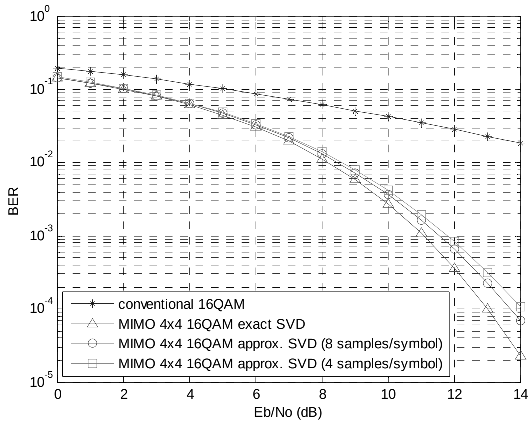

Fig. 3. BER as a function of the E_b/N_o for MIMO with exact SVD and CORDIC approximation compared to conventional system for 16QAM an n_T=n_R=4..

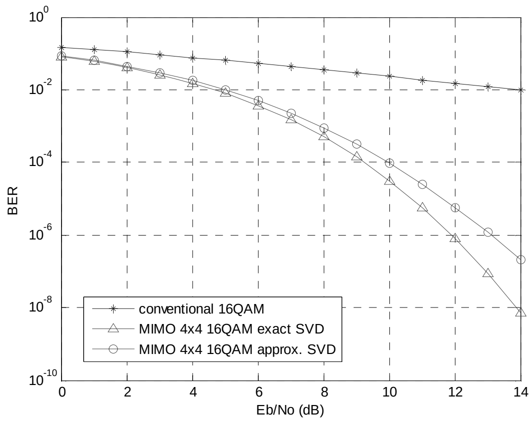

Fig. 4. BER as a function of the E_b/N_o for 4QAM an n_T=n_R=4.

VII. CONCLUSIONS

This paper has analyzed the implementation issues of the singular value decomposition (SVD) for modulation-mode and power assignment in MIMO systems on FPGA devices.

The implementation of Jacobi rotation for the SVD to optimize the system performance requires complex and time consuming operations that have been analyzed and a series of approximations have been proposed in order to diminish the computational load without large system performance degradation. Along the bibliography we find that authors propose different approximation techniques and implementation schemes for this goal.

This paper has shown that the use of approximations to diminish the computational complexity of the stated problem is feasible with a moderate degradation of the system behaviour with large implementation advantages.

Although current FPGA devices incorporate powerful computing units which incorporate multipliers, they increase the system cost, power consume and the number of gates required for the exact implementation of the SVD. Adoption of approximated solution decreases all this figures significantly. Pipelining and parallelism provides the possibility to further increase the system performance with larger throughputs.

The CORDIC algorithm is optimal for applications requiring large word lengths for the inputs when performance is not a restriction. When performance is a must the CORDIC algorithm can be implemented using parallelism and pipeline stages. Using CORDIC we can reduce the number of gates required to implement the algorithms.

The use of devices incorporating arithmetical units with multipliers (like digital signal processors or general purpose processors) can simplify the design task avoiding some of the simplifications described above and getting a more accurate calculation. Nevertheless, as shown below, the performance using approximations seems to be acceptable.

ACKNOWLEDGMENT

We acknowledge the Spanish *Ministerio de Industria* for the support of this work through different research projects: FIT-330101-2007-8, FIT-350201-2006-5, FIT-350100-2007-83.

REFERENCES

[1] A. Ahrens, C. Lange and C. Benavente-Peces, ” Modulation-Mode and Power Assignment in SVD- and GMD-assisted MIMO Systems”, 7th IASTED International Conference on Communications Systems and Networks (CSN), Palma de Mallorca (Spain), 01.-03. September 2008, submitted.

[2] R. P. Brent, F. T. Luk, C. Van Loan, “Computation of the Singular value Decomposition Using Mesh-connected Processors,”Journal of VLSI and Computer Systems, vol. 1, no. 3, pp. 242-270, 1985.

[3] Golub, G.H., and Van Loan, C.F., “Matrix Computations”, The John Hopkins University Press, Baltimore and London, 1989.

[4] Golub, G.H., and Kahan, W., “Calculating the singular value and pseudo-inverse of a matrix”, SIAM Journal, num. anal. ser. B 2, 205-224, , 1965.

[5] Hestenes, M.R., “Inversion of matrices by biorthogonalization and related results”, Journal of the Society for Industrial and Applied Mathematics”, Vol. 6, No. 1, pp. 51-90, Mar. 1958..

[6] J.E. Volder, “The CORDIC Trigonometric Computing Technique”, IRE Trans. Electronic Computers, vol. EC, no. 8, 1959, pp. 330–334.

[7] J. S. Walther, “A unified algorithm for elementary functions”, in “AFIPS Spring Joint Computer Conference”, vol. 38, pp. 379-85, 1971.

[8] R. Andraka,“ A survey of CORDIC algorithms for FPGA based computers”, in “Proceedings of the International Symposium on Field Programmable Gate Arrays”, Monterey, CA, USA, Feb. 1998, pp.191–200.

[10] J. Götze, “On the parallel implementation of Jacobi and Kogbetliantz algorithms”, SIAM Journal Science Computing, vol. 15, 1331-1348, 1994.

[11] C. Benavente-Peces, A. Ahrens, L. Arriero-Encinas and C. Lange Andreas AhrensM. J. Schulte and James E. Stine, ”Implementation analysis of SVD for modulation-mode and power assignment in MIMO systems” , 7th IASTED International Conference on Communications Systems and Networks (CSN), Palma de Mallorca (Spain), 01.-03. September 2008, submitted..